# ST(E) fan

Numer 5-6/93
Cena 30 000 zł
+ dwie dyskietki 40 000 zł.

**Pismo użytkowników komputerów Atari ST(E), TT i Falcon**

## POV Raytracer
## Trzeci wymiar grafiki komputerowej




## Atari Jaguar
## Sensacja!

news 3

rafał komorowski:
jaguar 7

krzysztof a. ziembik:
nie tylko myszka 9

tomasz dajczak:
emulator vt 52 10

robert wróblewski:
tips & tricks 10

rafał komorowski:
long live calamus 11

rafał komorowski:
stefan w holandii 12

dariusz paździor:
persistance of vision 14

marcin winiarski:
cywilizacja dla cywili 13

witold warczak:
szara skrzynka cz.3 18

midisterstwo:
martin midian mc key:
wstęp 23

martin midian mckey:
muzyka obrazu i słowa 24

martin midian mckey:
dnlt 25

krzysztof a. ziembik:
słów kilka o grafice i animacji cz.2 6

przemek kobel:
f/x 4,5 28

dariusz pilarczyk:
automaty komórkowe 29

witold warczak:
tos od środka cz.4 31

tomasz talkowski:
lynx 35

tomasz dajczak:
sound merlin 36

spis public domain 37

rafał komorowski:
dyskoteka stefana 39


# VAT INCLUDED

*Witajcie Kochani!*

*Numer „ST$^E$fana i dyskietki", który trzymacie w ręku jest pierwszym etapem pierestrojki, którą uchwaliliśmy w redakcji. Jest to także pierwszy numer wydany po wprowadzeniu nowego podatku od towarów i usług. Jak z dumą przyznajemy — cena nie uległa zmianie, co postaramy się utrzymać jak najdłużej.*

*Na czym polegają te rewolucyjne zmiany w naszym czasopiśmie? Po pierwsze zdecydowaliśmy, niestety niezgodnie z życzeniami Czytelników, że ostatecznie będziemy się ukazywać w cyklu dwumiesięcznym. Nasze dotychczasowe doświadczenia na tym polu zmuszają nas do stwierdzenia, że tak będzie lepiej... „ST$^E$fan" będzie miał natomiast podwójną objętość, w stosunku do tego, co sobie wcześniej planowaliśmy, dołączane do niego zaś będą dwie dyskietki. W przyszłości nie jest wykluczone zwiększenie objętości lub powrót do cyklu miesięcznego.*

*Po drugie, po wstępnej analizie ankiet nadesłanych przez Czytelników (patrz poprzedni numer) wprowadzamy nowy, stały dział: MIDISTerstwo. Komu jest on przeznaczony i o czym będzie traktował pozostawiamy do samodzielnego odgadnięcia przez Czytelnika. Niecierpliwych kierujemy na stronę 23.*

*Po trzecie będziemy się starali więcej pisać o rozrywce i edukacji, W CZYM LICZYMY NA POMOC CZYTELNIKÓW. Od następnego numeru zwiększy się także ilość opisów oprogramowania oraz recenzji nowych aplikacji. W numerze znajdziecie krótki opis konsoli do gier Lynx — dziecka Atari do niedawna u nas nieznanego. Ostatnio przeżywa ono swój renesans w postaci konsoli Lynx II, sprzedawanej obecnie także w Polsce. Wszystko co nosi znaczek Atari nie jest nam obce, dlatego będziemy więcej uwagi poświęcać temu produktowi, skoro inne pisma nie wypełniły dotychczas luki tu pozostawionej.*

*Drugi etap zmian — od następnego numeru — objawi się także w postaci zmiany graficznego oblicza „ST$^E$fana". Niestety, zmian oblicza poligraficznego nie możemy na razie obiecać... Wszystko zależy od nakładu pisma, ten zaś od sprawności dystrybucji, ta z kolei od liczby kupujących nas Czytelników.*

*Ale, ale! Zmiany i reformacje dotknęły nie tylko nasze czasopismo, ale przede wszystkim rynek Atari i sytuację firmy na świecie. Doniesienia są tak rewelacyjne, że nawet polskie gazety piszące o pecetach, takie jak „PC Kurier" odnotowały to w swoich „Newsach"! Co prawda przeinaczyli chłopcy co nieco, ale niech im tam będzie...*

*Chodzi i nową, fantastyczną konsolę do gier, lansowaną wspólnie przez Atari i... IBM! Nosi ona nazwę Jaguar, a więcej na ten temat znajdziecie w „Newsach" i na stronie 7. Drugą ważną wiadomością jest oficjalne stanowisko Atari na temat przyszłości i planów firmy. Jak deklaruje Jack Tramiel, Atari zamierza ponownie podbić masowy rynek konsumpcyjny. Lynx, Jaguar oraz Falcon są podstawowymi produktami przeznaczonymi właśnie na ten sektor. Natomiast rynek półprofesjonalny i profesjonalny Atari pozostawia firmom trzecim, które wspomagać będą użytkowników swoimi produktami i dodatkowymi peryferiami do komputerów ST/TT/Falcon. Oczywiście nie oznacza to całkowitego odwrócenia się od dotychczasowych użytkowników wykorzystujących swój sprzęt do poważniejszych zajęć. Toż sami calamusiarze rozdeptaliby siedzibę firmy Atari na proch! Wyjątkiem jest oczywiście także rynek MIDI, który zawsze miał silne wsparcie ze strony Atari i tak pozostanie. Dlatego wszyscy ci spośród Was, którzy oczekują na Falcona „035" lub 040 nie muszą załamywać rąk (są tacy! wiemy to z ankiet!). Dobrą wiadomością dla Was jest obietnica firmy, że do końca roku pojawi się w sprzedaży Falcon w obudowie desktop. Ten nowy komputer nie będzie bynajmniej starą „zerotrzydziestką" usadzoną do innego opakowania!*

*Dowodem na to, że firmy trzecie zaczynają forsownie wspomagać Atari w rozwoju jest klon (!) „tetetki", którego produkcję ma rozpocząć firma Ge-Soft. Obecna w wielu państwach europejskich firma Compo Software, wraz z kilkoma innymi dystrybutorami niemieckimi przejęła sprzedaż MultiTOS-u oraz SPEEDO GDOS-u, wraz z coraz pokaźniejszą biblioteką fontów m.in. firmy Bitstream.*

*Wszystko to wskazuje wyraźnie, Panie i Panowie, że nastały dla nas lepsze czasy, że Atari przezwyciężyło ostatecznie impas, w którym znajdowało się od paru lat. Górą nasi! Życzę Wam wszystkim ogromnego Falcona (w jakiej chcecie obudowie?), MultiTOS-u zainstalowanego na bardzo dużym dysku twardym i stu megabajtów najwspanialszych demosów*

*Wasz Rednacz*

Rafał Komorowski


ST$^E$-fan

**ST$^E$fan i dyskietka**, rok 2, numer 5–6, wrzesień'93.
Cena 30 000 zł,
z dwoma dyskietkami 70 000 zł, wysyłka 12 000 zł.
Nakład 5000 egzemplarzy.



**Redaguje zespół w składzie:**
Mateusz Karpow, Przemysław Kobel, Rafał Komorowski (red.nacz.), Oskar Kornfeld (fotografie), Robert Kowal, Daniel Michalik, Witold Warczak, Piotr „Gandalf" Wika, Krzysztof A. Ziembik.
**Stali współpracownicy:**
Michał Głębowski, Marek Kłodnicki, Piotr Miernikiewicz, Dariusz Paździor, Tomek Talkowski.
**Adres redakcji:**
ul. Trzemeska 12, pok.412, 53-679 Wrocław,
Tel/fax (071) 556460 ($11^{00}$–$17^{00}$).
**Adres do korespondencji:**
skrytka pocztowa 2180, 53-675 Wrocław 44.
**Projekt i DTP**: Mateusz Karpow.
**Polskie litery:** © Stefan Szczypka.
**Druk:** Kaster s.c. i Zakład Usług Poligraf., Wrocław, ul.Sudecka 95/97.



Liczymy na pomoc Czytelników w redagowaniu następnych numerów ST$^E$fan-a. W związku z tym zalecamy przyjęcie następujących standardów opracowań:

1.Co do tekstu:

— musi być napisany pod 1st Word Plus-em (chyba, że autor używa komputera zgodnego z IBM PC, wtedy pisze w zwykłym kodzie ASCII, polskie litery w standardzie Mazovii 152),

— prosimy nie dzielić wyrazów (włączona funkcja przerzucania całych słów),

— **każdy** nowy akapit musi zaczynać się tabulatorem!

— można używać tylko krojów: zwykły, kursywa, bold i bold kursywa.

2.Preferujemy następujące formaty rysunków P?1, P?2, P?3, PAC, IMG, ewentualnie TIF, GIF, XGA, TGA, LBM.

3.Moduły muzyczne akceptowane przez „CD Playera" (format tfu, tfu AMIGI), sample — format dowolny.

4.Programy w językach: GFA Basic (koniecznie plik *.LST), Pure C, Turbo C (pliki *.C i *.H) i Assemblerze (najlepiej MAS 68K).

5.Animacje w formatach: SEQ (Cyber Paint), TPA, POV, IIL (InShape), IFF, DLT.

**Wszystkie materiały prosimy przysyłać na dyskietkach (podpisanych)** — zwrot gwarantowany pod warunkiem podania adresu.

Z góry dziękujemy
i czekamy na listy
Redakcja

## Intermedia'93

W dniach 9–12 września we wrocławskiej Hali Ludowej odbędą się targi muzyczne Intermedia'93. Do odwiedzenia swoich stoisk zapraszają firmy Atar system (Atari, Emagic) oraz Polsound (Yamaha, Tascam, Steinberg). Targi zgromadzą ponad 100 wystawców z branży muzycznej. Pierwsze dwa dni (czwartek, piątek) przeznaczone są dla przedstawicieli firm, muzyków, itp. Pozostałe dwa dni — dla publiczności. Wstęp kosztuje ok. 60 000 zł w godz. 10–18. Serdecznie zapraszamy!

## Atari+IBM=Jaguar

Atari Corp. oraz IBM-Charlotte zawarły kontrakt na produkcję konsoli do gier nowej generacji — Jaguar. Kontrakt opiewa na pół miliarda dolarów i przewiduje, że zaprojektowane w laboratoriach Atari urządzenie będzie produkowane przez zakłady IBM w miejscowości Charlotte, USA. Wstępne dane techniczne Jaguara powalają z nóg: 64-bitowy procesor o architekturze RISC, do tego DSP, cena detaliczna: 200 (słownie: dwieście) dolarów. Szczegóły w artykule na stronie 7.

## Nowe gry na Falcona030

Jak dowiadujemy się z ST Formatu (nr 6/93) niedługo mają pojawić się następne gry przeznaczone wyłącznie dla Falcona. Pierwsza z nich to stary hit z konsoli Atari Lynx. Humans jest oparta na pomyśle Lemmingów i polega na przeprowadzeniu grupki jaskiniowców przez grotę i zbieraniu w tym czasie różnych przedmiotów.

Następną grą jest Raiden produkcji Imagitec z Wielkiej Brytanii. Jest to strzelanka na wysokim poziomie, polegająca na niszczeniu Twoim samolotem czołgów i dział wroga. Grafika oparta jest o pionowy przesuw ekranu... nad obozem nieprzyjaciela.

Steel Talons (Koveos, Wielka Brytania) to trójwymiarowa symulacja lotu śmigłowcem z elementami gry zręcznościowej. Twoim zadaniem jest niszczenie naziemnych celów przeciwnika i unikanie kontaktu z jego rakietami. Lot odbywa się oczywiście nad trójwymiarowym krajobrazem pola walki. Demonstracyjna wersja gry wyglądała na nieco wolną (jak na Falcona), ale producent obiecuje, że wersja ostateczna będzie znacznie, znacznie szybsza.

Być może znacie tę grę z automatów. W Road Riot 4WD (Images, Wielka Brytania) zasiadasz za kierownicą przerażająco szybkiego samochodu (choć znów nie bardzo to było widać w wersji demonstracyjnej). Jedynym Twoim zadaniem jest wyeliminowanie innych kierowców z wyścigu i powrót do domu z nagrodą. Gra zawiera w sobie wiele zdigitalizowanych obrazów na przykład panienek w bikini gratulujących Ci wygranej oraz dużo szybko skalowanej grafiki, co stwarza duże wrażenie realności.

Wiele z nowych gier na Falcona030, tak jak np. Llamazap Jeffa Mintera, korzysta z zaprojektowanego przez niego właśnie nowego 21-przyciskowego joysticka, który jest podłączony do gniazdka joysticka analogowego. Rozróżnia on osiem kierunków, posiada trzy przyciski fire oraz klawiaturę (podobną nieco do tej w niektórych telefonach) umożliwiającą szybki wybór opcji gry.

Wszystkie te gry mają dopiero się pojawić. Teraz istnieją tylko ich wersje demonstracyjne, które jak zapewniają producenci, nie ukazują pełnych ich możliwości. Zapowiadane wersje końcowe mają być o wiele lepsze... Natomiast francuska firma Silmarils od razu zabrała się do roboty. Ishar: Legend of the Fortress jest grą przygodową typu RPG (role-playing). Jej wersja na Falcona030 jest o wiele szybsza od tej na ST. O wiele ciekawsza grafika i dźwięk dodatkowo potęgują przeżycia. Jednak za bardzo przypomina ona wersję z peceta, więc można się domyślić, że Silmarils dokonało po prostu (na szybko) stosunkowo prostej konwersji z PC. O tym, co naprawdę potrafi Falcon będziesz się mógł dowiedzieć podobno z gry Ishar 2, której specjalną wersję na Falcona zapowiadało Silmarils na koniec maja.

# Nowości NEWS

Inne nowe gry zapowiadane na Falcona: Street Wariors (Excelsior, Francja) — klon Street Fighter 2, Unknown Ocean (Excelsior, Francja) — przygodowa, Alone in the Dark (Infogrames, Francja) — hit na pececie, Suriya (Lankhor, Francja) — przygodowa z syntezą mowy, Black Sect (Lankhor, Francja) — przygodowa, Vroom II (Lankhor, Francja) — wyścigi (do czterech graczy na ekranie).

## Devpac 3

W sprzedaży pojawił się ostatnio pakiet asembler i debuger HiSoft Devpac 3. Przedostatnia wersja, Devpac 2, cieszy (-ła) się ogromną popularnością wśród programistów. Prawdopodobnie -ła, z tego względu, iż wersja 3 zdobywa coraz więcej zwolenników. Devpac 3 przeznaczony jest dla wszystkich komputerów Atari wyposażonych w procesor 680x0 ($ST/ST^E/TT$/Falcon030). Potrafi także korzystać z koprocesora matematycznego. Proces asemblacji jest około 40% szybszy niż w poprzedniej wersji.

## ACC




Grupa Illusions zorganizuje ogólnopolskie Copy Party o nazwie „Atari Cracow Conference". Impreza przewidziana jest na koniec grudnia'93. Planowane miejsce — jak sama nazwa wskazuje — Kraków. Czas trwania: 2 dni. Przypuszczalna cena uczestnictwa 50 000 zł od osoby. W programie przewidziano Grafix Competition, Music Comp., Demo Comp. (jeżeli oczywiście zostaną zgłoszone jakieś prace). Ludzie do roboty!!! Zorganizowane będą również I otwarte ogólnopolskie mistrzostwa w grze Kick Off 2 na ST. Szczegółowych informacji na temat ACC szukajcie w czasopismach lub magazynach dyskowych o ST, na przełomie października i listopada.

Organizatorzy zastrzegają sobie możliwość odwołania imprezy bez podania przyczyny oraz ew. zmiany miejsca, w którym miałaby się ona odbyć. Telefon kontaktowy: Illusions (012) 76 34 17 (Wojtek, po $21^{00}$).

## Notator Logic 1.5

Firma Emagic (dawny C-LAB) wprowadziła do sprzedaży wersję 1.5 swojego programu Notator Logic. W ramach dużej akcji promocyjnej dotychczasowi posiadacze programów tej firmy mogą stać się posiadaczami Logica po korzystnych cenach. Przypomnijmy, że wspomniana firma jest producentem oprogramowania muzycznego (MIDI). Kolejne odmiany Notatora to: Alpha, SL oraz właśnie Logic. Różnią się one od siebie zaawansowaniem i ilością posiadanych funkcji oraz, oczywiście, ceną. Użytkownikom innego produktu tej firmy — Creatora SL — także umożliwiono odpowiedni upgrade na Notatora. Dystrybutorem produktów firmy Emagic w Polsce jest Atar System.

Atar System

ul.Trzemeska 12, p.412

53-679 Wrocław

tel/fax (071) 55-64-60

## Atari w sieci Novell

Firma Comtex Computersysteme z Niemiec umożliwiła podłączenie systemów Atari do sieci Novell. Podłączenie do wszystkich modeli Atari odbywa się poprzez szybki port DMA (ACSI), instalacja oprogramowania sprowadza się do skopiowania dwóch programów do katalogu AUTO na dysku startowym. Komputery Atari mogą bez problemów pracować jako końcówki w sieci, do której podłączone są nie tylko pecety, ale także komputery NeXT. Można wyobrazić sobie sieć, w której tylko serwer jest komputerem PC, wszystkie końcówki zaś to Atari.

Niestety cena takiego podłączenia do sieci nie jest niska: sprzęt wraz z oprogramowaniem kosztuje 998 DM za stanowisko.

Comtex Computersysteme

Gitterweg 3

7801 Bollschweil

Niemcy

## Biodata oferuje nową sieć — BioNet

Firma Biodata, znana z kilku rozwiązań sieciowych dla Atari przedstawiła sieciowy system BioNet 100 pozwalający zbudować sieć Ethernet w oparciu o komputery DOS, UNIX, Apple i Atari. W przypadku Atari dostępne są interfejsy odpowiednio wykorzystujące możliwości poszczególnych modeli, a więc złącze DMA w $ST/ST^E$, złącze VME w Mega $ST^E$/TT oraz SCSI w Falconie030.

Biodata GmbH

Burg Lichtenfels

3559 Lichtenfels

Niemcy

## Co nowego w DMC?

Na rynku pojawił się Calamus NT — nasz stary znajomy obecnie może pracować na stacji graficznej Olivetti obsługiwanej przez nowy system operacyjny firmy Microsoft, 32-bitowy Windows NT. Szczegóły i komentarze tego doniosłego acz kontrowersyjnego wydarzenia znajdziecie w osobnym artykule „Long live Calamus" wewnątrz numeru (strona 11).

Pojawił się także kolejny update do Calamusa SL dla Atari (czerwiec'93), jak zapowiada DMC ostatni darmowy dla dotychczasowych użytkowników. Nie starzeje się także Calamus S, młodszy, tańszy brat wersji SL, bo doczekał się właśnie wersji 2.0 wzbogaconej o wiele nowości, m.in. niektóre moduły ze wspomnianego najnowszego SL. Szczegóły o tych dwóch nowych wersjach programu także znajdziecie we wspomnianym artykule.

## Matrix MatGraph TC 1212

Nowa karta graficzna firmy Matrix znalazła już sobie w Polsce popularne określenie „dwanaście dwanaście". To potężne urządzenie przeznaczone jest dla Atari TT, przeważnie do profesjonalnego zastosowania w DTP i przetwarzaniu obrazów. Montowana jest w złączu VME, wyposażona w 8 MB szybkiej pamięci VRAM i drugie 8 MB DRAM i procesor graficzny TMS 34020. Osiągane rozdzielczości wprowadzają użytkownika na nowy poziom prosto ze stacji graficznych: 1280 x 1024 w pełnym, 24-bitowym kolorze. Przy ośmiu bitach (256 kolorów) karta jest w stanie generować wirtualnie obraz o wielkości 4096 x 2048 punktów. TC 1212 ma standardowo digitizer, który pozwala na digitalizację ze źródła wideo obrazu w formacie 768 x 576 w pełnym kolorze. Dodatkowy genlock wraz z encoderem pozwala na przegrywanie obrazów lub sekwencji obrazów z karty na magnetowid.

Matrix Daten Systeme GmbH

Talstraße 16

7155 Oppenweiler

Niemcy

## Procesor 68040 dla TT

Szwajcarska firma Medusa Computer System oferuje do sprzedaży płytę z procesorem

MC68040 przeznaczoną dla TT. Nowy procesor pracuje z częstotliwością 60 MHz, całość wyposażona jest w 8 MB pamięci Fast-RAM, obudowę tower wraz z zasilaczem i kosztuje 3050 DM. Na wyposażeniu znajduje się oprogramowanie emulujące nie zaimplementowane w 68040 funkcje koprocesora zmiennoprzecinkowego. Istnieje możliwość zakupienia samej płyty bez obudowy i pamięci, co pozwala wykorzystać posiadane już zasoby TT. Ceny poszczególnych konfiguracji dostępne są u producenta.

Fredi Aschwanden
Medusa Computer System
Postfach 3
CH-8610 Uster
Szwajcaria

## Klony Atari?

Taką rewelacyjną wiadomość przynosi firma GE-Soft, która wzięła sprawę w swoje ręce i postanowiła „wyręczyć" Atari w rozwijaniu profesjonalnego sprzętu do DTP (zgodnie z najnowszą strategią firmy Atari, o czym mowa w artykule wstępnym do tego numeru). Sklonowana ma zostać oczywiście „tetetka", robocza nazwa projektu brzmi Eagle. W obudowie pecetologicznej lub tower ma zmieścić się płyta główna zawierająca procesor 68030 taktowany 50 MHz. Uwagę należy zwrócić na fakt, że taką częstotliwością popędzany ma być nie tylko procesor ale i cała płyta, co wpływa ogromnie na szybkość pracy całego komputera.

Na dodatkowej płycie znajdują się typowe dla Atari złącza, np.: MIDI, ROM-port. Oprócz tego miejsce w obudowie znajdą dwa złącza VME (np. na dodatkową kartę graficzną) oraz pecetologiczne złącze ISA, taktowane marnymi 6.25 MHz (taka już jest jego natura...). Jakie może być zastosowanie takiego złącza w Atari pozostaje na razie sprawą otwartą. Istnieje możliwość wymiany procesora na 040, czyli coś takiego, jak w pecetach „overdrive ready". Prawdopodobnym jest możliwość zainstalowania DSP, co zbliża ten klon do Falcona. Pamięć zbudowana jest w oparciu o moduły SIMM, do rozbudowania TT-RAM-u nie potrzeba dodatkowych kart rozszerzeń (tak zwane ekspandery do „tetetki"). Maksymalna pojemność tej pamięci wynosi 64 MB (przy użyciu modułów 4 MB) lub 256 MB (gdy użyjemy SIMM-ów 16 MB). Jako ST-RAM użyta ma być szybka pamięć VRAM, standardowe wyposażenie przewiduje 4 MB, maksymalnie 14 MB. Obsługiwane rozdzielczości (bez kart graficznych) zredukowano o niepotrzebne w zastosowaniach DTP, a rozbudowano o te bardziej przydatne. Tak więc z rozdzielczości ST została tylko wysoka, „tetetkowe" tryby to TT-medium (640 x 480/16) no i TT-high (1280 x 960 mono). Dodatkowy tryb ma umożliwiać 256 kolorów w rozdzielczości 640 x 480, ale tu mogą nastąpić jeszcze pewne zmiany. Zastosowane mają być standardowe klawiatury ze świata PC. Przewidywana cena nowego komputera nie jest jeszcze ustalona.

GE-Soft
Landgrafenstr. 37–39
53842 Troisdorf-Oberlar
Niemcy

## Inkaust 2.1

Polski program do edycji i korekty ortograficznej tekstów Inkaust, którego producentem jest firma ELEN, pojawił się w nowej wersji. Przeznaczony jest on zasadniczo do współpracy z Calamusem i obecnie posiada obszerniejszy słownik ortograficzny będący głównym atutem systemu. Wprowadzono około 6,5 tys. nowych rdzeni wyrazowych, usunięto również szereg błędów. Prócz tego w programie poprawiono kilka błędów związanych z dzieleniem wyrazów (np. przedrostki OB, PRZE, PRZY i PÓŁ), błędów w obsłudze okna MAKRODEFINICJE, itp. Wprowadzono nową opcję „Dziel po pierwszej literze". Zniesione zostały także pewne ograniczenia słownika, jakie występowały w wersji 2.0. Dystrybutorem programu Inkaust jest Atar System.

Atar System
ul.Trzemeska 12, p.412
53-679 Wrocław
tel/fax (071) 55-64-60

## Nowy Chagall

W sprzedaży jest kolejna wersja programu graficznego Chagall firmy Trade iT. Pojawiły się pierwsze funkcje do obsługi ścieżek wektorowych i możliwość pracy z maskami 8-bitowymi. Bardzo widoczna staje się modularna koncepcja Chagalla, podobnie jak w Calamusie. Producent już zapowiada nowe moduły, co pozwoli przyszłym użytkownikom na odpowiednie „skomponowanie" programu i zakup tych, które są potrzebne w wykonywanych pracach, bez konieczności wydawania pieniędzy na nieużyteczne funkcje.

Wydaje się, że ten produkt wysforował się szybko do czołówki rastrowych programów graficznych dla Atari. Jeśli odniesie sukces na rynku to można liczyć na jego dalszy dynamiczny rozwój. Obecnie dostępny jest w dwóch wersjach: za 399 DM otrzymujemy program pracujący maksymalnie w 256 odcieniach szarości, za 699 DM wersję kolorową do 24 bitów (16 mln. kolorów).

Trade iT
Arheilgerweg 6
6101 Roßdorf/Hessen
Niemcy




*Nowy Chagall*

## Platon 2.3

Program do edycji płytek drukowanych firmy VHF-Computer bardzo często gości na łamach naszych NEWS-ów z powodu coraz to nowszych wersji. Obecnie dostępna wersja 2.3 przynosi następujące nowości: interaktywny autorouter pracujący w całości wektorowo, operujący nieco oszczędniej pamięcią komputera. Pojawiły się także biblioteki elementów TTL, CMOS oraz OPAMP. Następne w przygotowaniu.

VHF-Computer
Daimlerstr. 13
7036 Schönaich
Niemcy

## Twarde dyski AT-BUS

Niemieckie firmy oferują od pewnego czasu interfejsy pozwalające podłączyć dyski AT-BUS (inaczej IDE) do komputerów Atari. Jak wiadomo dyski tego standardu mają wolniejszą transmisję a ich konstrukcja nie pozwala na uzyskanie takich pojemności jak dyski SCSI. Złącze AT-BUS nie da się także zastosować do większej ilości jednostek (maksymalnie dwie, w porównaniu do siedmiu jednostek SCSI) ani do podłączenia innych urządzeń niż dyski — np. skanery, drukarki, CD-ROM-y. Ich atutem jest jednakże cena i rozpowszechnienie w świecie pecetów, a co za tym idzie — ogólnodostępność. Jeśli więc ktoś chce mieć tańszy lecz gorszy dysk rodem z PC — nie ma problemu. Oferowane obecnie interfejsy często zawierają także możliwość zainstalowania przy okazji nowej wersji systemu operacyjnego (przeważnie jest to TOS 2.06) — tak jest w przypadku produktów firm Gellerman oraz Hard & Soft. Oczywiście usprawnienie to nie dotyczy Falcona030, w którym kontroler AT-BUS zamontowany jest standardowo.

Przy okazji informujemy, że w numerze 9/93 niemieckiego czasopisma „c't" przedstawiono schemat sterownika AT BUS dla Atari ST, przeznaczony do samodzielnego wykonania.

## Kontroler SCSI w kablu

Jeśli ktoś chce pozostać jednak przy dysku SCSI a posiada Atari ze złączem ACSI (ST, Mega ST, $ST^E$, Mega $ST^E$, TT) to potrzebuje dokupić kontroler SCSI. Ostatnio montuje się je bezpośrednio w kablu połączeniowym pomiędzy komputerem a dyskiem, co pozwala w prosty sposób przyłączyć dowolny dysk SCSI nie przeznaczony do komputerów Atari. Kontroler taki oferuje np. firma Heyer & Neumann pod nazwą hdpLINK. Obsługuje on sygnał parzystości, co pozwala podłączyć napędy bez wyłączanego bitu parzystości (np. niektóre modele Quantum). Całość, wraz z oprogramowaniem zarządzającym dyskiem no i kablem kosztuje około 200 DM.

Heyer & Neumann GbR
Hansmannstr. 19
5100 Aachen
Niemcy

## Nowy napęd CD-ROM firmy Sony

Firma Ventas Technologies oferuje nową stację CD-ROM na bazie napędu CDU-561 firmy Sony. Stacja ma możliwości czytania dysków Kodak-Photo-CD (tzw. multisessioning), normalnych CD-ROM-ów standardu X/A oraz CD-I. Zwiększona została prędkość rotacji dysku, przez co osiągnięto mniejszy czas dostępu i dwukrotnie większą szybkość transferu danych (300 KB/s zamiast 150 KB/s). Napęd wyposażony jest w bufor wielkości 256 KB i posiada interfejs SCSI rozszerzony o nowy tryb SCSI II znany np. z Falcona. Stacja firmy Ventas umożliwia także odczytywanie normalnych kompaktów audio i przesyłanie ich zawartości w postaci cyfrowej do komputera!

Ventas Technologies
Brüsseler Str. 21
50674 Köln
Niemcy

## Emulator peceta dla Falcona

Stało się, Heim-Verlag wypuścił urządzenie FalconSpeed, które służy do emulacji peceta na Falconie. Osiągi nie są rewelacyjne, ale pozwalają na zaspokojenie większości potrzeb pecetowca z Windowsami 3.1 włącznie. Konkretnie: procesor 286, 16 MHz, grafika jak to Falcon: VGA, SVGA. Wszystkich tych, którzy skrzywili się przeczytawszy „286" śpieszymy poinformować, że średnie osiągi w codziennej pracy opisywanego urządzenia porównywalne są z 386 20 MHz, głównie dzięki szybkiej grafice Falcona. Wykazują to np. testy wykonane pod Windowsami.

Urządzenie podłączane jest pod szynę rozszerzeń Falcona, nie wymaga lutowania i kosztuje 498 DM. Można sądzić, że to dopiero pierwszy krok w temacie upecetowiania Falcona, następnym powinien być emulator 386 i 486.

Heim-Verlag
Heidelberger Landstraße 194
W-6100 Darmstadt 13
Niemcy

## Rozszerzenia pamięci dla Falcona

Jeśli ktoś chciałby mieć 14 MB pamięci ST-RAM w Falconie, a jest już posiadaczem „ptaszka" z 4 MB to może zamienić posiadaną pamięć na nową. Firma Layout-Service-Kiel oferuje kartę 14 MB, identyczną rozmiarami z poprzednią (czyli nie ma problemów z ciasną obudową komputera), zbudowaną z energooszczędnych kości (nie powodujących nadmiernego rozgrzewania się układu).

Inne rozszerzenie, pod nazwą falconWINGS, oferuje firma Heyer & Neumann. Dostępne są wersje 8 MB (cena 888 DM), 12 MB (1249 DM) oraz 14 MB (1549 DM), przy czym ta ostatnia wersja dostarczana jest wraz z rozszerzeniem graficznym ScreenBlaster, o którym piszemy niżej. Dodatkowo producent odkupuje poprzednio zainstalowaną w Falconie płytkę pamięci 4 MB za 50 DM.

Firma Richter Distributor także oferuje pamięć do Falcona — wykonaną przy użyciu zwykłych pakietów SIMM. Cena (bez kości pamięci) wynosi 248 DM. Niestety nie wiadomo nam jaka jest maksymalna pojemność tego rozszerzenia i który rodzaj pamięci wchodzi w rachubę: ST-RAM czy TT-RAM.

Layout-Service-Kiel
Eckernförder Str. 83
24116 Kiel 1
Niemcy

Heyer & Neumann GbR
Hansmannstr. 19
5100 Aachen
Niemcy

# Nowości NEWS

Richter Distributor
Hagener Str. 65
58285 Gevelsberg
Niemcy

## 32 MHz dla Falcona

Richter Distributor jest też producentem przyśpieszacza do naszego ukochanego „flakona". Oferowany jest 32-megahercowy dopalacz do tego komputera. Dostępny jest w dwóch wersjach: za 448 DM przyśpieszamy posiadany już procesor, natomiast za 598 DM otrzymujemy układ z nową Motorolą w wersji 32 MHz. Całość montowana w gnieździe rozszerzeń nie wymaga lutowania.

Ge-Soft, firma znana z kart rozszerzeń pamięci dla Atari TT także oferuje rozbudowę Falcona do 32 MHz. Mighty Sonic 32 kosztuje około 500 DM, wyposażony jest w nowy procesor 32 MHz, natomiast wersja kosztująca 1200 DM umożliwia także rozbudowę komputera o pamięć typu TT-RAM (inaczej Fast-RAM) zgodną z analogiczną pamięcią w Atari TT. Dotychczasowe rozkazy systemowe odwołujące się do TT-RAM-u pozostały w Falconie niezmienione, a więc wszystkie programy, które umieją korzystać z tego rodzaju pamięci (praktycznie wszystkie „znające" TT", czyli powiedzmy od 1991 roku większość nowych produktów) będą mogły korzystać z tej pamięci pracując na Falconie. Maksymalna pojemność rozbudowanej pamięci zależy od rodzaju użytych modułów SIMM i przy użyciu kości 4 MB wynosi 32 MB, natomiast przy użyciu SIMM-ów 16 MB wynosi 128 MB. Jak dotychczas wszystkie opisywane rozszerzenia Falcona także i to nie wymaga lutowania przy instalacji i jak wynika z opisu producenta jest przelotowe, tzn. w gnieździe tym można zainstalować coś jeszcze. Wielkość opisywanego rozszerzenia (w wersji z TT-RAM-em) nie umożliwia jednak zainstalowania całości w normalnej obudowie Falcona030, co oczywiście i tak nie ma sensu. Dlatego aby użyć Mighty Sonic 32 z rozbudowaną pamięcią potrzebujemy przełożyć komputer do innej obudowy, np. od „tetetki", co pokazaliśmy pokrótce w poprzednim numerze, lub do obudowy typu tower, co oferuje już od pewnego czasu kilka firm niemieckich.

Tym posunięciem możemy na bazie Falcona skonstruować system np. do DTP o mocy obliczeniowej równej TT (nie licząc zastosowań DSP!), pamięci do 128 MB i rozbudowywalnym systemie pamięci masowych SCSI, od większego dysku twardego począwszy, przez dyski wymienne SyQuest, a skończywszy na magnetooptycznych stacjach zapisywalnych 128 MB i 650 MB.

Richter Distributor
Hagener Str. 65
58285 Gevelsberg
Niemcy

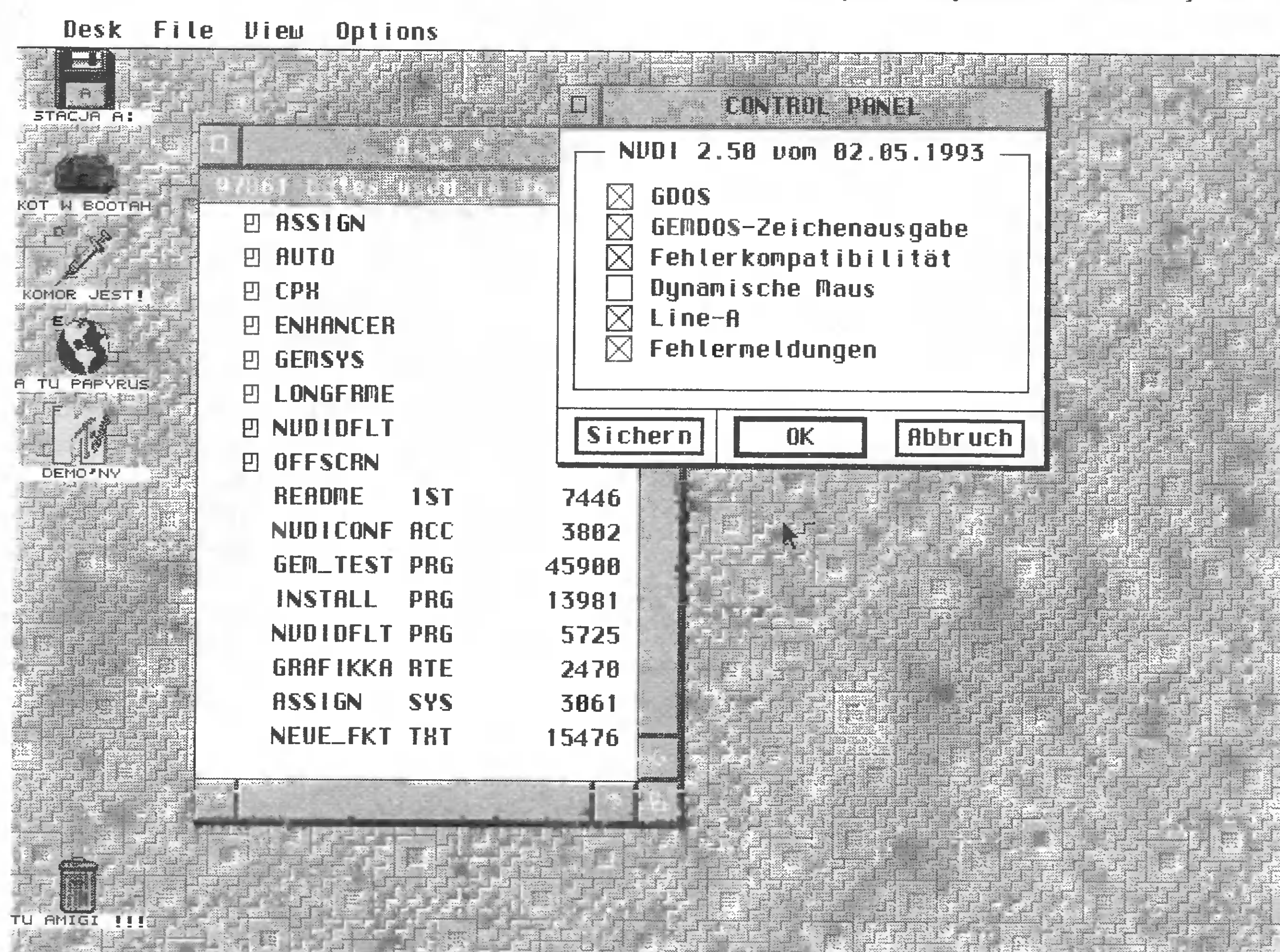
GE-Soft
Landgrafenstr. 37–39
53842 Troisdorf-Oberlar
Niemcy

## NVDI dla Falcona030

Firma Bela Soft, producent legendarnego już programowego „przyśpieszacza" graficznego dla Atari przygotowała wersję 2.50 swojego sztandarowego produktu. Nowe NVDI pracuje na Falconie i daje niesamowitego kopa temu komputerowi. Kto twierdzi, że „goły" Falcon ma szybką grafikę niech popatrzy na Falcona z zainstalowanym NVDI — już „na oko" widać różnicę. Doświadczyliśmy tego w redakcji, kto raz spróbował pracy z NVDI ten nie będzie chciał powrócić do poprzedniego stanu. Ubocznym efektem pracy NVDI na Falconie jest... pojawienie się polskich znaków — większość dotychczasowych akcesoriów instalujących polskie znaki nie pracowało zbyt skutecznie na tym komputerze.

## Rozszerzenia graficzne do F030

Pierwszym rozszerzeniem możliwości graficznych nowego komputera Atari (poza licznymi programowymi) jest ScreenBlaster firmy OverScan. Jest to sprzętowo-programowe rozwiązanie pozwalające na poprawienie rozdzielczości nawet do 300 procent (!) w zależności od posiadanego monitora. Na poczciwym SM 144 uzyskać moż-




*NVDI 2.50 pracuje we wszystkich rozdzielczościach Falcona030. To ładne tło jest zasługą programu Deskpic*

na 880 x 496 punktów, na monitorze VGA 880 x 600 non-interlace nie stanowi problemu, wyższe rozdzielczości do 1280 x 624 są dostępne w trybie interlace, natomiast monitor klasy multisync to 1024 x 656/16 kolorów non-interlaced, 1152 x 832/16 interlaced, ale przy 87 Hz częstotliwości pionowej, czyli migotania prawie nie ma! Na deser mamy 800 x 608/256 kolorów. Najwyższe częstotliwości w pionie dochodzą w niektórych trybach do 97 Hz, a wtedy interlace nie straszny... Ponadto oprogramowanie umożliwia pracowanie w rozdzielczościach wirtualnych, tzn. na ekranie widoczna jest tylko część całego obszaru roboczego. Przesuwając mysz w pobliże obrzeża ekranu następuje automatyczny przesuw w zadanym kierunku. Pozwala to ułożyć sobie wygodnie na wirtualnym ekranie np. 5000 x 5000 punktów wszystkie potrzebne okna bez potrzeby ciągłego ich odkrywania czy otwierania. Pozwala to zaoszczędzić wiele czasu. Mamy ponadto szybszy dostęp i lepszy „widok" na opracowywany dokument czy pisany tekst programu. Cena ScreenBlastera jest niewygórowana — 149 DM. Można go także dostać w bardzo korzystnym pakiecie z opisywanym wyżej NVDI za 199 DM. Zestaw ScreenBlaster+NVDI to wybuchowe połączenie pozwalające osiągnąć dużą wygodę i szybkość pracy. Polecamy bez zastrzeżeń każdemu, kto chce wykorzystać Falcona do poważniejszej pracy.

Następnym rozszerzeniem jest czysto programowe (!!!) rozwiązanie firmy Neumann-Seidel o wszystko mówiącej nazwie High-Videlity. Najprzydatniejszy tryb to chyba 832 x 608 n-i (non-interlaced) przy częstotliwości pionowej do 70 Hz. Jeśli chodzi o rozdzielczości i (interlace) to mamy do dyspozycji np. 1280 x 960 przy 44 Hz w pionie. Cena równie rewelacyjna: 49 DM.

Ostatnią ofertą jest równie wybuchowe urządzenie o nazwie BlowUp030 wyprodukowane przez firmę Acher, Eberl, Seibert GbR. Oferowane w trzech wersjach, od czysto programowej za 29 DM (!!!), przez pośrednią wersję sprzętową za 99 DM, skończywszy na najlepszej wersji za 129 DM (również sprzętowej). Osiągane maksymalne rozdzielczości to: dla monitora SM 124 896 x 480 n-i, dla VGA 880 x 480 n-i oraz 1152 x 912 i, SGVA zaoferuje nam solidny standard: 800 x 608 n-i przy 72 Hz. Multisync to 880 x 608 n-i oraz 1280 x 1024 i. Wyobrażacie sobie? Monochromatyczna rozdzielczość TT-high, osiąganą na profesjonalnym 19-calowym monitorze, niegdyś duma atarowców jest teraz dostępna na nowym, domowym komputerze Atari, ale do tego w szesnastu kolorach!!! Co prawda w trybie interlace, ale za to przy 69 Hz.

BlowUP030 wyróżnia się możliwością płynnej zmiany parametrów pracy. Oznacza to, że do wyboru mamy nie tylko ustalony, ograniczony zestaw gotowych rozdzielczości, ale sami możemy — do ustalonych granic — regulować ilość punktów w poziomie, pionie i częstotliwości odchylania. Przypomina to generatory rozdzielczości do wielu prawdziwych kart graficznych dla Atari ST/TT. A trzeba pamiętać, że wszystkie opisywane urządzenia nie są pełnoprawnymi kartami graficznymi (z własną pamięcią, procesorem itp.) ale tylko rozszerzeniami „podrasowującymi" standardowe osiągi Falcona! Zadziwiające jest, jaką elastycznością odznacza się procesor wizji w Falconie, jak niewiele jego możliwości odkryła przed nami firma Atari. BlowUP030 także udostępnia wirtualne rozdzielczości, wspomniane przy opisie ScreenBlastera.

Oto adresy producentów tych rozszerzeń (wszystkie w Niemczech):

OverScan,
Säntisstr. 166
D-12277 Berlin

Neumann-Seidel GbR
Zwikauerstr. 4
56075 Koblenz

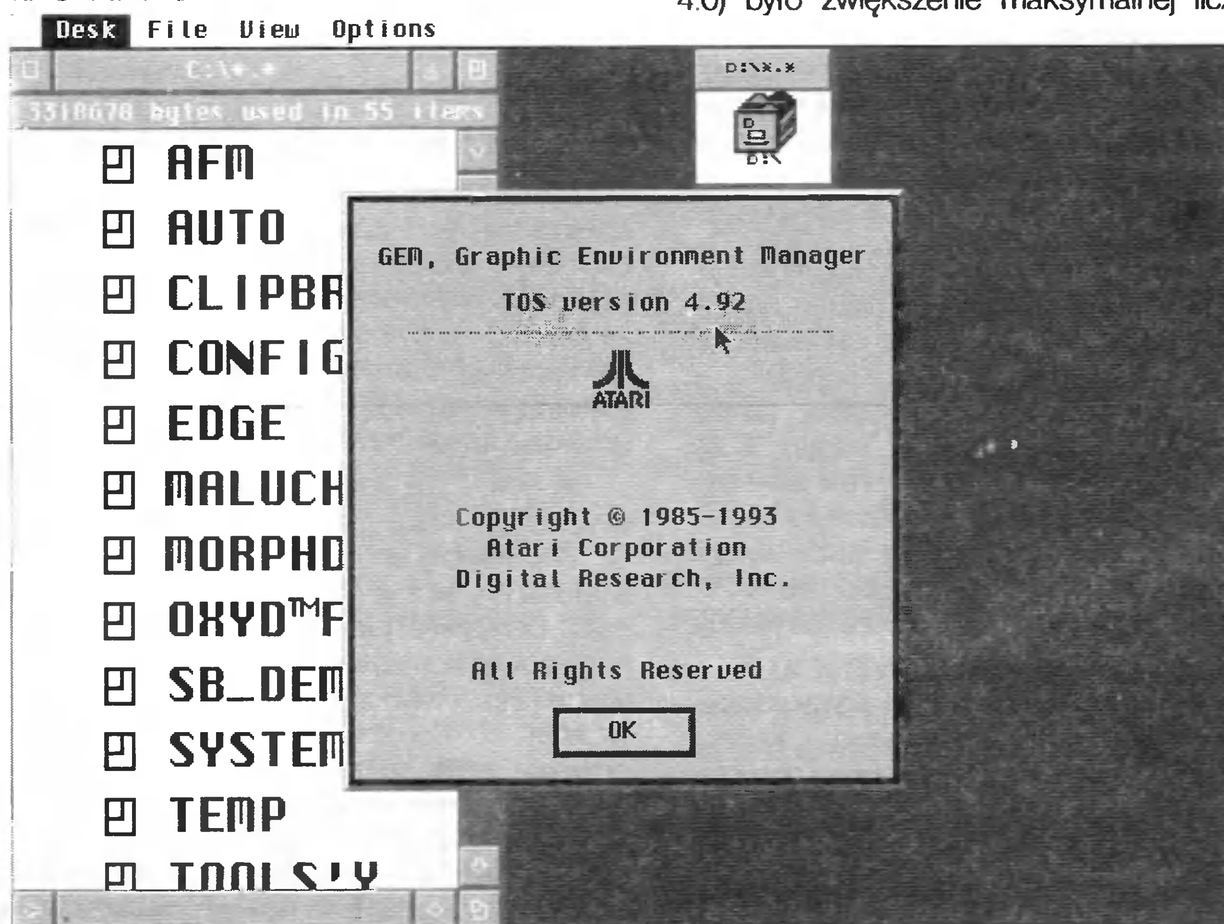
Nowości
# NEWS

Acher, Eberl, Seibert GbR
Eslarner Str. 34
81549 München

wersji 1.32 (program kategorii shareware do Falcona, służący do nagrywania muzyki na twardy dysk).

À propos desktopu w nowym TOS-ie: posiada on możliwość wyboru fontu używanego w oknach z zawartością dysku. Dostępne są wszystkie zainstalowane w systemie fonty, w dowolnych wielkościach, włącznie z wektorowym Speedo GDOS-em.

Od pewnego czasu nowe wersje systemu operacyjnego pojawiają się tak często, że nie nadążamy z ich opisywaniem. W Falconie instalowane były kolejno wersje: 4.0, potem 4.01, następnie 4.02, obecnie zaś jest tam instalowana wersja 4.04. Jedną z ważnych zmian wprowadzonych „gdzieś około" TOS-u 4.01 (a może już w 4.0) było zwiększenie maksymalnej liczby okien




*Tak będzie wyglądał przyszły TOS 5.0, którego premiera przewidziana jest na koniec roku. Po prawej stronie na górze widać zikonizowane okno. Po lewej zaś stronie otwarte okno, w którym widać nowy przycisk i zmienione fonty.*

## Falcon w obudowie desktop...

Z europejskiej siedziby Atari (Vianen, Holandia) otrzymaliśmy informację o nowych produktach, jakie firma Atari planuje sprzedawać jeszcze w tym roku. Po pierwsze — Falcon w obudowie desktop. Ta wiadomość ucieszy wielu z Was, którzy od dawna czekają na Sokoła w takiej właśnie obudowie. Nowy Falcon ma mieć w pełni 32-bitową magistralę, co przyśpieszy działanie systemu, prawdopodobnie większą częstotliwość zegara, być może nowe tryby graficzne.

## ...z nowym TOS-em i Multi-TOS-em.

Do tego w fazie zaawansowanych testów beta znajduje się kolejna wersja systemu operacyjnego, prawdopodobnie 5.0. Dzięki modemowym staraniom Piotra Wiki posiadamy beta-wersję oznaczoną 4.92. Główne nowości to: osobny, znajdujący się na dysku/dyskietce plik RSC zawierający zasoby desktopu i systemu, co umożliwi łatwą i bezbolesną zmianę wersji językowej, osobne pliki konfiguracyjne zawierające definicję klawiatury i fontów systemowych (nareszcie!). No i, szanowni Państwo, w oknie pojawi się nowy przycisk: iconizer, obok wypełniacza (ang. fuller). Oznacza to, że otwarte okno można zikonizować, tzn. zmniejszyć do minimalnych rozmiarów, by trzymać je gdzieś pod ręką. Nie wszystkie „dotychczasowe" okna można zikonizować, gdyż operacja ta musi być obsługiwana przez programy. Dotychczasowe aplikacje nie mają więc w swoich oknach nowego przycisku. Jednakże nie tylko nowy desktop obsługuje ikonizację. Mimo, że nowy TOS nie jest jeszcze ogólnie dostępny — są już pierwsze aplikacje wykorzystujące tą możliwość, np. WinRec w

możliwych do otwarcia w systemie z siedmiu do... no właśnie, dokładnie nie wiemy ilu. Nowy desktop pozwala ich otworzyć osiem, ale do tego dochodzą jeszcze okna akcesoriów, nie udało nam się dokładnie policzyć. W programie Interface w systemie 4.04 można ich otworzyć przynajmniej dziesięć.

Cały system 5.0 jest bazowany na rdzeniu z MultiTOS-u i w wielu sytuacjach zachowuje się właśnie jak ten ostatni, oczywiście z wyjątkiem uruchamiania kilku aplikacji. Przykładem tego jest możliwość ładowania akcesoriów podczas pracy (nie tylko podczas startu systemu). Premiera nowego TOS-u ma być połączona z ukazaniem się odpowiedniej do tego wersji MultiTOS-u, obsługującego wszystkie wymienione cechy.

## DIGIT II

Firma Galactic, znana z produkcji samplerów i oprogramowania dla nich oferuje obecnie użytkownikom Falcona030 rodzinę programów DIGIT II. Pierwszy z nich — DIGIT II Studio — jest już dostępny w sprzedaży i umożliwia samplowanie dźwięku do pamięci RAM lub na twardy dysk, za pomocą wbudowanych w komputer przetworników. Zaimplementowano wszystkie podstawowe operacje edycyjne, takie jak wycinanie i wklejanie bloków, edycję, czy pracę z lupą. Możliwe jest także odtworzenie zaprojektowanej sekwencji dźwięków (taki prosty sekwencer) lub granie sampli od tyłu. Program wymaga TOS-u przynajmniej 4.02. Przewidywana cena DIGIT II Studio wynosi 150 DM.

Galactic
Julienstr. 7
45130 Essen 1
Niemcy

Rafał Komorowski

# Jaguar – świat gier XXI wieku

**Wiadomość o nowej konsoli do gier, która jest wspólnym przedsięwzięciem firm Atari oraz IBM zelektryzowała wszystkich zainteresowanych. Ujawnione dane techniczne wskazują, że jest to naprawdę konsola rodem z dwudziestego pierwszego wieku, przebijająca wszystko to, co dotychczas zaprezentowano na masowym rynku gier komputerowych. Cena dwustu dolarów, jakie trzeba wydać, by stać się posiadaczem Jaguara wydaje się nierealnie niska.**

## Kości zostały rzucone

28 czerwiec 1993 — Sunnyvale, CA, USA — Atari Corporation ogłasza, że podpisany został kontrakt z oddziałem IBM Corporation, firmą Charlotte, North Carolina, USA. Kontrakt dotyczy produkcji konsoli Atari Jaguar, nowego, 64-bitowego, multimedialnego systemu do gier komputerowych. Kilkuletni kontrakt z firmą IBM opiewa na pół miliarda dolarów.

Atari Jaguar, którego produkcja ma odbywać się w Stanach Zjednoczonych, jest interaktywnym systemem multimedialnym, wyróżniającym się grafiką 24-bitową (ponad 16 milionów kolorów). System ten jest w stanie tworzyć na ekranie trójwymiarowe, cieniowane płaszczyzny, tworzące „realny świat" — w czasie rzeczywistym. 32-bitowy port rozszerzeń pozwoli na przyszłe podłączenie do sieci kablowej i telefonicznej, port procesora sygnałowego umożliwi proste podłączenie modemu i zewnętrznych urządzeń audio. Jaguar będzie mógł być także podłączony do zewnętrznego odtwarzacza płyt CD (CD ROM), o podwójnej szybkości pracy.

*„Ten system jest zdecydowanie powiewem przyszłości"* powiedział Sam Tramiel, dyrektor firmy Atari. *„Jaguar będzie prezentował takie zaawansowanie technologiczne, że zdecydowaliśmy się współpracować z przodującą firmą komputerową, o której wiemy, że potrafi produkować wysokiej jakości wyroby, które my zaprojektujemy".*

Oddział IBM — Charlotte — który przez ostatnie 15 lat produkował i rozwijał produkty tylko dla potrzeb koncernu-giganta, ostatnio rozpoczął współpracę z innymi przedsiębiorstwami, by zaspokoić ich potrzeby technologiczne. Projekt Atari Jaguar jest jednym z pierwszych prób IBM-a wejścia na masowy rynek popularnych zastosowań elektroniki.

*„To wspaniała okazja pracować z Atari i ich nowym systemem"* powiedział Herbert L. Watkins, z firmy IBM Charlotte. *„Wszyscy widzą IBM-a jako wytwórcę systemów elektronicznej informacji oraz technologii, teraz pokażemy, że jesteśmy w stanie wyprodukować skomplikowany produkt przeznaczony na rynek zastosowań domowych".*

Oprócz montowania Jaguara IBM będzie także odpowiedzialny za zaopatrzenie w podzespoły, kontrolę jakości, pakowanie i dystrybucję. Jaguar bazowany jest na zaprojektowanym przez Atari 64-bitowym procesorze RISC współpracującym z procesorem sygnałowym (DSP) także projektu Atari (nie mylić z innym DSP firmy Motorola, który „siedzi" w Falconie!), odpowiedzialnym za wysokiej jakości, stereofoniczny dźwięk CD. 64-bitowa konsola do gier prześciga czterokrotnie obecne na rynku systemy Nintendo, czy Sega, oparte na procesorach 16-bitowych.

Sierpień'93 — na zamkniętym pokazie dla producentów gier i dystrybutorów po raz pierwszy pokazano wygląd przyszłej konsoli i kilka gier, które bądź są na ukończeniu, bądź już przygotowane do produkcji.

Jesienią tego roku Jaguar będzie dostępny w pierwszej, sondażowej serii, na rynku Nowego Jorku i San Francisco. Na terenie USA wielka sprzedaż spodziewana jest na początku przyszłego roku. Ogłoszona cena Jaguara powala z nóg — dwieście dolarów...

## Nowa generacja

Konsole do gier znane są od lat. Produkuje je wiele firm na świecie, od małych, przenośnych, z monochromatycznym ekranem LCD bez podświetlania (np. Gameboy firmy Nintendo), poprzez nieco większe, ale przenośne systemy z kolorowym ekranem LCD i podświetlaniem (Lynx firmy Atari). Dalej plasują się proste konsole stacjonarne rodem z połowy lat osiemdziesiątych, podłączane do telewizorów (Pegasus, różne produkty firm Nintendo i Sega). W ostatnich latach pojawiły się jednakże nowe konstrukcje, o wyższej klasie (i cenie), oparte na 16-bitowych procesorach, z lepszym dźwiękiem i bardziej kolorową grafiką.

Jako najnowsza awangarda prezentowany jest przyszły system firmy 3DO, którego produkcję na podjął Panasonic. Oparty jest on o 32-bitowy procesor ARM60 i wyposażony w DSP, jednakże zapowiedziana cena 500–700 dolarów wydaje się być zbyt wygórowana, mimo, że wliczono w nią CD-ROM.

W takiej sytuacji prezentowana przez Atari konsola Jaguar staje się awangardą awangardy i towarem, którego moc obliczeniowa jest nieporównywalna z niczym dostępnym na tym rynku!

Gry o 24-bitowej trójwymiarowej grafice, dźwięku CD i skomplikowanych przekształceniach (np. morphing) obrazu są możliwe. Rozkompresowywane w czasie rzeczywistym dane, w tym także ruchome obrazy pozwalają zaprojektować grę o niespotykanej dotąd ilości grafiki i dźwięku, bez martwienia się o dziesiątki megabajtów potrzebnych do pomieszczenia tego wszystkiego.

*„Kiedy statek kosmiczny obraca się wokół księżyca, osoba spacerująca na ulicy skręca za róg — każdy obiekt takiej sceny jest ukazywany jako trójwymiarowy obraz, cieniowany i pokryty skomplikowaną powierzchnią (ang. texture). To naprawdę fantastyczne!"* stwierdza Sam Tramiel.

Do Jaguara przewidziane są kasetki *(ang. cardridge)* MegaCard mieszczące sześć megabajtów danych, co po zastosowaniu specjalnych metod kompresji (Atari wspomina o technologii Cine Pak) pozwala na umieszczenie do 50 MB danych!!! Za dodatkowe 200 dolarów Atari będzie w przyszłym roku oferować napęd CD o podwójnej szybkości pracy, pozwalający na odtwarzanie zwykłych płyt CD, CD+G, Kodak PhotoCD no i oczywiście kompaktów przeznaczonych specjalnie dla Jaguara.

## Z wyglądu...

Obudowa Jaguara nie została jeszcze publicznie zaprezentowana, niemniej krążą już o niej plotki w modemowych sieciach komputerowych. Sterowanie w grach Jaguara odbywać się za pomocą specjalnego joysticka pod nazwą Power Pad, wyposażonego dodatkowo w około dziesięć klawiszy spełniających różne role, w zależności od gry. Power Pad oraz jedna z dostępnych już gier (nie wiadomo jeszcze która) wchodzi w skład standardowego wyposażenia Jaguara.

Konsola wyposażona jest w dwa megabajty RAM-u, slot na kasetki MegaCard służący jednocześnie jako wejście na wspomniany, opcjonalny CD-ROM.

Dodatkowo Jaguar posiada wyjście ComLynx, które pozwala połączyć ze sobą kilka Lynxów i Jaguarów! Pozwala to na prowadzenie gry w kilka osób. Oznacza to, że niektóre gry z Lynxa będą dostępne także dla Jaguara (i być może na odwrót), co jest interesującą możliwością. Nie ma niestety przewidzianej możliwości używania kasetek przeznaczonych dla Lynxa na Jaguarze.

## Co w środku

Konstruktorzy firmy Atari od trzech lat opracowywali założenia, architekturę systemu i poszczególne układy Jaguara. Całości przyświecała idea, by produkt upakować w jak najmniejszej ilości układów scalonych, co pozwoli obniżyć koszty, zmniejszyć wymiary, a zwiększyć niezawodność i szybkość pracy. Dlatego w dwóch fizycznych scalakach upchnięto kilka logicznych jednostek, stanowiących o sile Jaguara. Reszta to kości pamięci, kilka układów TTL i niewiele ponadto.

Oto główne dane techniczne Jaguara:

* 64-bitowa wielozadaniowa architektura RISC.
* 64-bitowa magistrala o wysokiej przepustowości (106.4 MB/s).
* Pięć procesorów: (nie oznacza to pięciu układów scalonych!)
  — 27-MIPS-owy *Graphic Processor* posiadający 4 kilo wewnętrznej, szybkiej pamięci static-RAM, powiązany blisko z blitterem, potrafi generować szeroką gamę efektów graficznych z cieniowaniem i rotacją włącznie. Programowalny.
  — Programowalny *Object Processor* mogący być podstawą wielu różnych architektur wideo. Może służyć jako zaawansowany kontroler duszków, przekształceń rastrowych, itp.
  — 27-MIPS-owe DSP wyposażone w 8 kilo static-RAM, obsługuje 16-bitowy dźwięk stereofoniczny.
  — Blitter o dużej mocy przetwarzania, wspomagający sprzętowo tzw.*Z-buffering* oraz *Goudraud shading*.
  — Poczciwa Motorola 68000 taktowana 13.3 MHz — jako kontroler ogólnego zastosowania.
* Realistyczny 32-bitowy kolor na telewizorze NTSC lub PAL.
* Rozdzielczość 720 x 576 (telewizor!).
* ROM *cardridge* o pojemności 6 MB (bez kompresji).
* 2 MB szybkiej pamięci RAM (tzw. *fast page-mode*).
* Opcjonalny napęd CD-ROM w cenie ok. 200 dolarów, dostępny w 1994 roku.
* ComLynx pozwalający na łączenie kilku Lynxów i Jaguarów.
* Dwa (rozszerzalne do większej ilości) wejścia na specjalne kontrolery: klawiatury, pistolety świetlne, myszy oraz inne cyfrowe i analogowe urządzenia.
* Wysokiej jakości, ergonomiczne *joypady* wyposażone w trzy przyciski*fire*, Pause, Option oraz 12-klawiszowy *keypad* do dowolnego zastosowania w grze. Do gier mogą być dołą-

czane nakładki, które po założeniu na tą klawiaturkę będą stanowić opis używanych klawiszy.

* Szybkie synchroniczne złącze szeregowe do podłączania modemu, telewizji kablowej i różnego rodzaju sieci.

W konsoli znajdują się dwa duże układy scalone, Tom i Jerry, które razem mieszczą w sobie cztery procesory Jaguara. Jerry zawiera DSP, 8 kilo static-RAM, blok dźwięku i kilka pomniejszych funkcji. Tom posiada kontroler pamięci DRAM, *Object Processor, Graphic Processor* wraz z jego pamięcią static-RAM oraz blitter. Motorola 68000 jest osobnym układem, wokół którego skupiona jest pamięć DRAM oraz obsługa kasetki oraz CD-ROM-u.

## O własnych siłach...

Inżynierowie Atari musieli włożyć wiele pracy i wysiłku, by zaprojektować układy nie tylko wystarczająco silne i szybkie, ale i nie wykraczające poza założone koszty produkcji. Szczególnie ważna okazała się przepustowość magistrali komputera. Richard Miller, wiceprezydent ds. badań i rozwoju wspomina: *„Grafika zabiera mnóstwo czasu na magistrali. To, co jest obecnie dostępne dla innych procesorów 64-bitowych, takich jak PowerPC to właśnie tyle, ile chcieliśmy osiągnąć. Zaprojektowaliśmy coś, co jest na technologicznym poziomie drogich 64-bitowych jednostek."*

Aby koszty nie były zbyt wysokie trzeba było z technologii krzemowej wycisnąć wszystko co możliwe. Układy Tom i Jerry produkowane będą przez jednego z czterech największych producentów na świecie (nie ujawniono jeszcze konkretnej nazwy firmy). Magistrala wewnętrzna komputera jest w pełni 64-bitowa.

Na początku inżynierowie poszukiwali odpowiednich procesorów i układów DSP pośród dostępnych na rynku, lecz nie znalazło się nic, co spełniałoby warunki zarówno co do szybkości pracy jak i kosztów. Nie pozostało nic innego, jak zrobić to wszystko samemu.

## ...i razem z innymi

Atari przy projektowaniu Jaguara pozostawało w ścisłej współpracy z kompanią Time Warner. To właśnie ta firma podpowiedziała projektantom zastosowanie wyjścia na kablową i telefoniczną sieć, która w przyszłości może okazać się bramą do świata gier sieciowych, toczonych przez wielu użytkowników oddalonych od siebie o setki kilometrów, połączonych jedynie telefonem...

Dotychczasowa produkcja komputerowa Atari odbywała się na Dalekim Wschodzie. Produkcja Jaguara będzie odbywać się w inny sposób. *„Aby produkować jedną z najbardziej skomplikowanych konsol na świecie potrzebowaliśmy kogoś, kto wie co to jest masowa produkcja wysoce skomplikowanych technologicznie wyrobów"* stwierdził Richard Miller *„IBM był naturalnym wyborem."*

## Powiew przyszłości

Efekty zastosowania tak nowoczesnych rozwiązań są zbijające z nóg. Szybkość grafiki jest porównywalna do szybkości średniej klasy unixowych stacji graficznych. 64-bitowy procesor technologii RISC jest w stanie nie tylko szybko budować zawartość ekranu, przeprowadzać skomplikowane efekty graficzne, ale także rozkompresowywać w czasie rzeczywistym animacje zapisane na kasetce z grą lub na CD ROM-ie.

## RISC-owy procesor Jaguara

Atari zapewnia, że Jaguar jest w stanie dokonać *renderingu* algorytmem Goudraud ponad 50 milionów punktów na sekundę. Procesor graficzny wraz z blitterem zapełniają ekran 850 milionami punktów w ciągu jednej sekundy. Przez magistralę Jaguara przepływać może strumień danych wielkości ponad 100 MB/s. To naprawdę niesamowite, szczególnie jeśli zna się osiągi innych konsol, nawet tak zaawansowanej konstrukcji, jaką ma być 3DO. Więcej szczegółowych liczb i faktów porównujących najważniejsze konsole obecne na rynku znajdziecie w ramce poniżej.

## Jaguar the king

Firma Atari, ojciec rynku gier komputerowych i producent znanych tytułów na wiele różnych automatów, konsol i komputerów, powraca na pozycję kreatywnej siły nadającej rynkowi kierunek i rozpęd. Rozpoczęcie sprzedaży rewolucyjnej multimedialnej platformy, jaką jest Jaguar, połączone będzie z agresywną polityką reklamową i promocyjną. Atari planuje sprzedać w tym roku 50 tysięcy Jaguarów na rynku Nowego Jorku i San Francisco. Przyszły rok ma oznaczać pół miliona w całych Stanach.

Prowadzone są prace nad przystosowaniem znanych gier-przebojów dla nowego systemu. Wersje dla Jaguara wyróżniają się trójwymiarową grafiką i fantastycznymi efektami niedostępnymi w pierwowzorach. Na zamkniętej prezentacji dla przyszłych dystrybutorów i producentów gier przedstawiono już gotowe lub prawie gotowe tytuły dla Jaguara. Ich krótki opis znajdziecie w następnym numerze. ❑

| | **Jaguar** | **3DO** | **SNES** | **GENESIS** |
|---|---|---|---|---|
| **Szerokość szyny** | 64 bity | 32 bity | 16 bitów | 16 bitów |
| **Szybkość renderingu i animacji** | 850 milionów punktów/sek. | 64 Miliony punktów/sek. | 1 Milin punktów/sek. | 1 Milion punktów/sek. |
| **Przepustowość szyny** | 106,4 MB/sek. | 60 MB/sek. | ? | ? |
| **Kolory** | 16,7 mln | 16,7 mln | 256 | 64 |
| **Grafika True Color** | Tak (32-bitowa) | Tak (24-bitowa) | Tak (16-bitowa) | Nie |
| **Procesory** | 5: GPU + DSP procesor do obieków blitter + 68000 | 4: ARM60 + DSP 2 procesory graficzne | 2: 65C816 DSP | 2: 68000 Z80 |
| **16-bitowy dźwięk jakości CD** | Tak | Tak | Nie | Nie |
| **Ilość MIPS** | 55 | ? | ? | ? |
| **Specjalizowana budowa (do 3D)** | Tak | Nie | Nie | Nie |
| **Architektura wieloprocesorowa** | Tak | ? | ? | ? |
| **Procesor do tworzenia obiektów** | Tak | Nie | Nie | Nie |
| **Wyjście S-Video** | Tak | Tak | Tak | Nie |
| **Modulator** | Tak | Tak | Tak | Tak |
| **Wyjście Composite** | Tak | Tak | Tak | ? |
| **Wyjście RGB** | Tak | ? | ? | Tak |
| **Rozdzielczość** | 720 x 576 | 640 x 480 | 512 x 448 | 320 x 224 |



Krzysztof A. Ziembik

# Nie tylko myszka

**Od wielu lat naczelną myślą projektantów komputerów jest maksymalne ułatwienie życia użytkownikom, tak, by jak najmniej czasu musieli poświęcać na nauczenie się obsługi i dojście do perfekcji. W dziedzinie oprogramowania przełomem w traktowaniu właścicieli komputerów był graficzny system operacyjny.**

Zwolniło to z pamiętania karkołomnych kombinacji klawiszy czy wydawania skomplikowanych komend.

W graficznym systemie operacyjnym wszystko widać i nie sposób się pogubić. Nie mógłby on jednak istnieć bez innego przełomowego osiągnięcia jakim jest myszka. Mysz to niepozorne pudełeczko, które jako pierwsze, dodatkowe urządzenie do komputera odrywało użytkownika od nieustannego klepania w klawiaturę. Z czasem myszom przybywało uszu, ich kształty były coraz bardziej obłe, przystosowane do kształtów ludzkiej dłoni. Później niektóre myszy odwrócono na grzbiet (powstawał kot czyli track-ball). Nie poprzestano jednak na maltretowaniu niewinnych zwierząt. Zaczęto wymyślać pióra świetlne, tabliczki graficzne, „laserowe" pistolety, myszo-pióra, itd., itp. I wszystko po to, żeby łatwiej porozumiewać się z komputerem. Jeden z takich przedmiotów chciałbym właśnie opisać.

## Tripad

Przedmiotem moich testów, przeprowadzonych w stosunkowo krótkim czasie i bez zaginionej w redakcji instrukcji, stał się system Tripad składający się z nietypowej tabliczki graficznej Robotron K6405 z zasilaczem i dwoma rodzajami wskaźników (zwanych też czujnikami) oraz programu instalującego Tripad 30. Pierwotnie główna część tego zestawu (czyli sprzęt) była przygotowany do współpracy z innym rodzajem komputerów (PC?) i przystosowanie do działania z ST i pochodnymi wymagało pewnych zmian i uzupełnień, które zostały wykonane przez firmę Tritec z Berlina. Najpierw więc kilka słów o sprzęcie, oczywiście z punktu widzenia użytkownika, a nie elektronika.

### Zasilacz

Zasilacz ma postać dużego, czarnego pudełka o wymiarach ok. 20 x 14 x 8 centymetrów i jest wykonany z solidnej blachy. W jednym z boków znajduje się wyłącznik, obok wychodzą z obudowy dwa przewody. Pierwszy, o długości ponad 1,5 metra służy do podłączenia do sieci, drugi, dobrze ponad metrowy dołącza się do interfejsu umieszczonego w porcie modemu komputera. Podsumowując walory użytkowe zasilacza trzeba powiedzieć, że jego gabaryty nie są szczytem wygody, ale rekompensują to dość długie kable, pozwalające go ustawić w znacznej odległości od komputera.

### Tabliczka

Tabliczka, oprócz górnej powierzchni koloru białego, jest szara, pasuje więc kolorystycznie do ST-ka. Spód jest wykonany z blachy, reszta jest z plastiku. Ma spore wymiary, ok. 49 na 38 centymetrów, jej grubość wynosi od 4 cm w tylnej części do 2 cm z przodu (jest lekko pochyła, tak jak klawiatura ST). Z tyłu obudowy znajdują się trzy rzeczy: gruby przewód z interfejsem podłączanym do portu RS-232 w ST (modem), wejście, do którego dołącza się wskaźnik (sensor) i przycisk zerujący (reset). Na wierzchniej stronie są cztery diody sygnalizujące pracę i pole robocze o wymiarach 31 na 21 cm, które przy standardowych parametrach jest odpowiednikiem ekranu. Obok, po lewej stronie są dodatkowe pola do ustawiania niektórych parametrów za pomocą wskaźnika.

### Wskaźnik 1

Wskaźnik pod nazwą Robotron K7702 jest małym (7 x 4,5 cm) i płaskim (1 cm) urządzeniem. Przypomina mysz, ale ma tylko jeden przycisk, a zamiast w kulkę jest wyposażony w celownik optyczny. Posiada kabel i to praktycznie tej samej długości co zwykła atarowska mysz, zakończony nietypową wtyczką (8 bolców) z dwoma zatrzaskami (pomysłowe). Wskaźnik podłącza się do tabliczki graficznej z tyłu jej obudowy.

### Wskaźnik 2

Drugi z wskaźników (czujników) nie posiada oznaczenia fabrycznego. Z wyglądu przypomina gruby pisak, z którego wychodzi kabel zakończony wtyczką. Kabel i wtyczka są oczywiście identyczne jak w pierwszym wskaźniku. Pisak ma zwykłą długość, 15 centymetrów, jest zrobiony z czarnego plastiku. Wyposażony jest w podobny mechanizm jak popularne „automatyczne" ołówki, ale zamiast rysika ma sztywny plastikowy drucik, który dociśnięty do powierzchni symuluje naciśnięcie lewego klawisza myszy.

### Program

Program instalujący Tripad 30 został napisany w firmie Tritec przez Markusa Zella. Jest to akcesorium, co poważnie ogranicza możliwości jego stosowania w niektórych przypadkach. Na dysku zajmuje jedynie 22 KB, ale po uruchomieniu aż 75 KB pamięci RAM. Z tego powodu używanie Tripada na ST z 0,5 MB jest utrudnione, wręcz niemożliwe, gdy program z którym chcemy go używać nie zostawia wiele wolnego RAM (Crack Art 1.25 nawet przy 1 MB nie da się uruchomić z powodu braku pamięci). Poza samym instalatorem na dysku są programiki źródłowe w dwóch językach programowania, pozwalające obsługiwać Tripad we własnych programach.

## W akcji

Połączenie wszystkiego do kupy nie jest skomplikowane. Również nietrudne jest uruchomienie instalatora z dyskietki dostarczanej razem z pozostałymi częściami systemu. Pierwsze próby działania, na biurku systemowym, wypadły pomyślnie. Pierwszym wskaźnikiem można jeździć nie tylko po powierzchni tabliczki, ale nawet do 3 centymetrów nad nią! Pozwala to na swobodne poczynania z kursorem na ekranie, nawet gdy przypadkowo oderwiemy wskaźnik od tabliczki. Można regulować między innymi prędkość przesuwu kursora i związaną z tym czułość wskaźników. Można także zmieniać wymiary powierzchni roboczej deklarując jej lewy, dolny i prawy, górny róg (a więc od 651 $cm^2$ do nawet 1 $cm^2$). Dzięki temu z tabliczką można poprawnie pracować w każdej rozdzielczości ST (trybów TT i Falcona nie testowałem). Pewną wadą jest to, że nie jest możliwe jednoczesne korzystanie z myszki i wskaźnika (chodzi o poruszanie, przyciski są równouprawnione).

Posługiwanie się obydwoma wskaźnikami zmusza do pewnej zmiany przyzwyczajeń, ponieważ sterowanie jest trochę inne niż myszą. Myszą można wszystko robić prawie w miejscu, przesuwać kursor niewielkimi skokami, podnosząc mysz i wracając do miejsca startu. W systemie Tripad trzeba się przyzwyczaić że pole robocze tabliczki graficznej jest kopią pola roboczego ekranu i nie da się skrócić toru ruchu wskaźnika. Poza tym jest różnica między wskaźnikami. Ten z celownikiem optycznym jest płynniejszy w działaniu, ale dość niewygodny dla ręki. Pisak jest wygodny (choć nie można zbytnio go pochylać, tak jak zwykły długopis czy ołówek), ale przy szybszym poruszaniu nie nadąża z odczytem pozycji i na ekranie mamy efekt skokowo przemieszczającego się kursora.

Do czego jednak wykorzystać to dość ciekawe urządzenie? Na pierwszy ogień poszły programy graficzne na niską rozdzielczość, z których korzystam najczęściej. Niestety, okazało się, że tak już przywykłem do posługiwania się przy rysowaniu myszą, iż korzystanie z tabliczki jest makabrą i przypomina pisanie lewą ręką przez praworęcznego osobnika. Oczywiście nie ma to znaczenia dla chcących posługiwać się Tripadem na stałe, mogą oni przecież poświęcić trochę czasu na przyzwyczajenie się. Przy okazji wyszła na wierzch pierwsza poważniejsza wada systemu, a właściwie wskaźników. Mają one tylko jeden przycisk, równoważny lewemu klawiszowi myszy, co w licznych programach (DEGAS Elite, ZZ-Rough, NEOchrome, Crack Art, Cyber Paint) jest niewystarczające. Pomocna jest wtedy mysz, posłusznie podłączona do ST i udostępniająca drugi, prawy klawisz. Są także kłopoty z programami nie całkiem GEM-owymi, gdyż program instalujący może działać niepoprawnie (np. Crack Art 1.0 czy Cyber Paint, w których instalator nie pojawia się w ACC) lub w ogóle nie działać (Deluxe Paint). W ten sposób traci się możliwość używania Tripada w kilku całkiem dobrych programach. Zauważyłem jeszcze inną dokuczliwą wadę. Przy korzystaniu z całego pola roboczego kabel wskaźnika często zaczepia o róg tabliczki (na szczęście łatwo da się to wyeliminować).

W drugiej części testu wypróbowałem programy rzadziej przeze mnie wykorzystywane, służące do tworzenia grafiki monochromatycznej, a także programy typu CAD (najogólniej mówiąc przeznaczone do projektowania brył). Tabliczką można się posłużyć do przerysowywania istniejących na papierze obrazów, zarówno czarnobiałych ilustracji, jak i schematów technicznych. W takim wypadku wystarczy włożyć papier między wskaźnik a tabliczkę. Znów powtarzają się problemy z brakiem drugiego przycisku, choć już nie w takim natężeniu. Okazuje się, że programy na mono i CAD nie za często potrzebują obu i wtedy wskaźnik jest „samowystarczalny". Poza tym w trybie mono więcej programów jest GEM-owych i z instalatorem nie ma kłopotów. Należy też wspomnieć, że niektóre programy (właśnie CAD) są dostarczane z własnymi sterownikami dla tabliczki graficznej i wtedy jej możliwości mogą być inaczej, najczęściej lepiej, wykorzystanie. Np. pole robocze tabliczki może stać się odpowiednikiem okienka, a nie całego ekranu jak jest standardowo. Wtedy przesuwanie wskaźnika spowoduje przesuwanie zawartości okna bez użycia suwaków!

System Tripad jest w obecnej postaci ciekawym rozwiązaniem i przy spełnieniu pewnych warunków nadającym się do pracy, szczególnie może być przydatny w programach do projektowania. Ma jednak dwa podstawowe błędy w założeniach (jeden przycisk w wskaźniku, instalator w ACC), które zawężają grono potencjalnych użytkowników. Pocieszeniem jest to, że producent Tripada za dodatkową opłatą dostarcza wskaźnik z dwoma, a nawet trzema przyciskami. Niestety nie wiem jakie są koszty takiej zamiany.

## No i co z tego?

No i to, że można używać nie tylko myszki. Warto popróbować innych rzeczy, może ktoś z zaskoczeniem stwierdzi, iż np. tabliczka graficzna lub pióro świetlne lepiej „leży mu w dłoni" i łatwiej mu będzie stworzyć arcydzieło graficzne, na które w ST$^E$fanie trwa przecież nieustający konkurs (patrz strona obok). A może ktoś zainspiruje się podanymi informacjami i skonstruuje jakieś nowe narzędzie, coś, co będzie działać i da się podłączyć do ST?

❑



Tomasz Dajczak

# Emulator VT52

Dawno, dawno temu, pewnego słonecznego dnia, pisałem prosty, aczkolwiek pożyteczny programik. Programik nie wykorzystywał możliwości GEM-u, obce mu były okienka oraz rozwijane menu. Wystarczyło mu wyświetlanie komunikatów na ekranie oraz możliwość wpisywania poleceń za pomocą klawiatury.

W końcu zetknąłem się z koniecznością wykorzystania zwykłego, prostokątnego kursora i umieszczenia go w pewnym miejscu ekranu. Po przeglądnięciu spisu treści książki „Atari Profibuch" znalazłem procedurę o obiecującej nazwie Cursconf (XBIOS 21). Można było za jej pomocą wywołać kursor, usunąć go z ekranu, a nawet zmienić częstotliwość jego migotania. „Brakuje tylko wodotrysku" pomyślałem. Wodotryskiem, jak się okazało chwilę później, był taki nieistotny szczegółik jak pozycjonowanie kursora na ekranie. Po piętnastu minutach poszukiwań funkcji przesuwającej niesforny kursor przypomniałem sobie o istnieniu czegoś takiego jak emulator VT52. Co jednak kryje się pod tą tajemniczą nazwą?

Do wyświetlania tekstu na ekranie najwygodniej jest użyć funkcji Cconws (GEMDOS 9). Powoduje ona pojawienie się na ekranie dowolnie długiego tekstu. Koniec tekstu jest rozpoznawany po bajcie równym zero, który musi się tam znajdować. Jednak w danych do wyświetlenia nie muszą się znajdować tylko litery i cyfry. Mogą tam być również tajemnicze sekwencje kodów, zaczynające się od ESC (ASCII 27). Po wywołaniu GEMDOS-u, zostaną one wykryte przez procedurkę zwaną emulatorem VT52 i zamienione na ruch kursora, kasowanie ekranu, czy inne pożyteczne rzeczy.

Czytelnicy ST^Efana, którzy posiadali kiedyś (lub posiadają) Atari 800 XL, stwierdzą zapewne, że na tym komputerze również zaimplementowano coś takiego. Za przykład podadzą dziwne znaczki, które wpisywane do tekstu w instrukcji PRINT Basica powodowały kasowanie ekranu, przeskok tabulacji czy ruch kursora. Sterujące sekwencje kodów występują również w drukarkach, gdzie kierują wyborem czcionki czy drukiem grafiki. Tak więc nie są one niczym niezwykłym. Aby je wykorzystać, potrzebna jest tylko ich lista i trochę dobrych chęci. A więc:

**ESC A — kursor w górę**

Kursor zostanie przeniesiony o wiersz do góry. Jeśli znajdował się on w górnej linii, nie zostaje przesunięty.

**ESC B — kursor w dół**

Kursor zostanie przesunięty o wiersz, w kierunku środka Ziemi. Jeśli znajdował się on w najniższej linii ekranu, pozostanie głuchy na namowy na spacer.

**ESC C — kursor naprzód**

Kursor przesunie się w kierunku prawego brzegu ekranu.

**ESC D — kursor wstecz**

Kursor poruszy się w kierunku lewego brzegu ekranu.

**ESC E — czyść ekran, kursor w pozycję wyjściową**

Nieodwracalnie kasuje zawartość ekranu, kursor zostaje przesunięty w lewy, górny róg ekranu.

**ESC H — kursor w pozycję wyjściową**

Powoduje przeskok kursora w lewy górny róg ekranu.

**ESC I — kursor w dół, przesunięcie ekranu**

Kursor zostanie przesunięty o wiersz w dół. Jeśli znajdował się w najniższej linii ekranu, zostanie utworzony nowy wiersz, a reszta ekranu przesunie się do góry.

**ESC J — kasowanie do końca strony**

Kasuje ekran od bieżącej pozycji kursora (włącznie) aż do końca strony.

**ESC K — kasowanie do końca wiersza**

Kasuje tekst od bieżącej pozycji kursora do końca wiersza.

**ESC L — wstaw wiersz**

Wstawia dodatkowy wiersz w bieżącej pozycji kursora. Kursor znajdzie się na początku nowej linii, zaś reszta ekranu przesunie się w dół.

**ESC M — kasuj wiersz**

Kasuje aktualny wiersz i przesuwa resztę ekranu w górę. Jeśli kursor znajdował się w najniższej linii ekranu, tekst zostanie skasowany, a kursor przesunięty do lewego marginesu.

**ESC Y — pozycjonuj kursor**

Pozycjonuje kursor na ekranie. Po tych dwóch kodach, powinny się znaleźć dwa bajty, określające pozycję Y oraz X (liczone od lewego górnego rogu ekranu, powiększone o 32).

**ESC b — ustaw kolor tekstu**

Ustawia kolor, w jakim wyświetlany będzie tekst. Po nim następuje bajt określający numer koloru.

**ESC c — ustaw kolor tła**

Podobnie jak ESC b, ale zmieniany jest kolor tła.

**ESC d — kasuj początek ekranu**

Kasuje tekst od lewego górnego rogu do bieżącej pozycji kursora.

**ESC e — włącz kursor**

Powoduje, że kursor staje się widoczny.

**ESC f — wyłącz kursor**

Kursor popada w nirwanę.

**ESC j — zapisz pozycję kursora**

Bieżąca pozycja kursora zostaje zapisana. Gwoli ścisłości napiszę, że przechowywana jest ona w rejestrze sav_row (WORD $4AC, dziesiętnie 1196). Nie jest to jednak oficjalnie udokumentowane.

**ESC k — odtwórz pozycję kursora**

Odtwarza zapisaną pozycję kursora. Działa jednorazowo (pozycja zostaje skasowana z pamięci).

**ESC l — kasuj bieżący wiersz**

Wiersz tekstu, w którym znajduje się kursor zostanie wymazany. Kursor wraca na lewy brzeg ekranu.

**ESC o — kasuj początek wiersza**

Tekst od początku wiersza do bieżącej pozycji kursora zostanie usunięty.

**ESC p — włącz tryb odwrócony**

Włącza tryb odwrócony (kolory tekstu i tła zostaną zamienione).

**ESC q — wyłącz tryb odwrócony**

Wyświetlanie tekstu wraca do starego trybu.

**ESC v — przejście na końcu wiersza**

W tym trybie, po osiągnięciu końca wiersza zostaje automatycznie utworzona następna linijka.

**ESC w — zatrzymanie na końcu wiersza**

W tym trybie, nowy wiersz zostanie utworzony dopiero po wysłaniu kodów Powrót Kursora (CR — ASC 13) i Wysunięcie Wiersza (LF — ASC 10).

Sądzę, że opis kodów kontrolnych przyda się domorosłym programistom, szczególnie tym nie posiadającym literatury lub nie znającym języków obcych. Mając te wiadomości można pokusić się o napisanie np. prostego edytora tekstu, ładnego i pamięciooszczędnego ekranu informacyjnego czy czegokolwiek innego. Jedynym ograniczeniem jest tutaj inwencja twórcy. Dla niedowiarków na dyskietce do ST^Efana załączam listing w asemblerze, wykorzystujący opisane tutaj cechy naszego poczciwego komputera. ❑

# TIPS & TRICKS

```
move.w #2,-(SP)
trap    #14
addq.l  #2,SP
lea     bsadr1(PC),A0
move.l  D0,(A0)+
move.l  #8*25*160,D0
add.l   D0,(A0)
move.l  cleartext(PC)
pea     #9,-(SP)
move.w  #1
trap
bs und farbe auf rot
```

Wiele osób (do których i ja się zaliczam) ma olbrzymie kłopoty z przedzieraniem się przez gąszcze komnat, hordy wrogich śmigłowców itd. I chyba dlatego całe rzesze fanatyków stukają w klawiatury, próbując znaleźć coś, co pozwoli przebrnąć drogę od startu do endu (oczywiście happy) bez straty jednego „życia" lub przy minimalnym zużyciu energii. Znajdując odpowiednie słowo lub kombinację klawiszy jesteśmy w stanie ukończyć grę, w której nie mogliśmy przejść pierwszego etapu. I to jest to. Z angielskiego nazywamy to tipsem. I można powiedzieć, iż Anglicy w dziedzinie wyszukiwania ich przodują. Wiele osób sądzi, że tipsy są to pewnego rodzaju furtki zostawione przez programistów, po to aby w każdej chwili mieć bezpośrednią możliwość sprawdzenia zachowania się programu w określonym miejscu w fazie uruchamiania. A więc szukajmy i chwalmy się nimi.

Na dobry początek przedstawiam troszkę tipsów z moje kolekcji (która obecnie liczy z 200 sztuk):

**Defender 2** — w czasie gry należy wpisać RAVEN, aby za pomocą N przechodzić kolejne poziomy.

**Dragon Breed** — w czasie gry należy wpisać IREM aby uzyskać nieśmiertelność.

**Exolon** — na tablicy najlepszych wyników należy wpisać AD ASTRA aby uzyskać nieśmiertelność.

**F 29 Retaliator** — aby mieć cały czas multum broni wpisz CIARAN na rejestracyjnym ekranie.

**Horror Zombies** — BOGEYMAN, aby uzyskać wieczne życie.

**Navy Seals** — WOZZIE na tablicy najlepszych wyników, przez H zatrzymaj grę i ESC aby przeskoczyć na następny poziom.

**Robocop 3** — trzymając Shift i wpisz THE DIDDY MEN, naciskając ESC dostajesz się na następny poziom.

**Sim City** — naciśnij CapsLock i wpisz FUND dostaniesz $10000.

**Swiv** — zatrzymaj grę i wpisz NCC 1701, nie działa to na niektórych wersjach pirackich.

**Terminator 2** — zatrzymaj grę, naciśnij wszystkie klawisze funkcyjne (F1, F2, F3, ...,F10), odpal grę przez Fire i naciśnij ESC aby przejść na następny poziom.

**BATMAN** — kiedy pojawi się ekran tytułowy wpisz MAJJJJJJ..., trzymaj tak długo J aż pojawi się napis „Cheat Mode" (lewy, górny róg), F10 pozwala na zmianę etapu.

**Titus the Fox** — kody do kilku poziomów: 8455, 1933, 8612, 9813, 5052, 2045.

**Bombuzal** — kody do kilku etapów: RATT, BIRD, RING, GIRL, LOCK, SAFE, TREE, MYTHI.

Na koniec ciekawostka: na tablicy najlepszych wyników w **Double Dragon** wpisz NEIL HARDING a usłyszysz cóż się sądzi o Amigach.

I to byłoby na tyle. Następna porcja za jakiś czas. Połamania joya

Robert Wróblewski



Rafał Komorowski

# Long live Calamus

**DMC, producent legendarnego Calamusa nie ustaje w wysiłkach zmierzających do dalszej popularyzacji tego systemu na rynku komputerowym. Świadczy o tym powstanie Calamusa w wersji przeznaczonej dla nowego systemu operacyjnego — Windows NT.**

Niektórzy użytkownicy Atari i komentatorzy podnieśli krzyk: to już koniec Atari, firma DMC opuszcza swoją dotychczasową platformę i szuka nowych dróg rozwoju na rynku PC! Zamieszanie było na tyle duże, że potrzeba było aż oficjalnej wypowiedzi przedstawiciela DMC na łamach niemieckiego „ST-Magazin", aby obniżyć temperaturę. Na szczęście wygląda na to, że pesymiści mylili się.

### Komu SL, komu NT?

Świadczy o tym kilka faktów. Po pierwsze liczba dotychczasowych (legalnych) użytkowników Calamusa jest na tyle duża, że nie sposób tak po prostu z nich zrezygnować. Przecież oni są potencjalnymi nabywcami nowych wersji programu, nowych modułów i towarzyszących Calamusowi programów, np. Outline Art. O zainteresowaniu i trosce firmy o dotychczasowych klientów świadczy wypuszczenie nowych wersji Calamusa SL oraz S. DMC zapowiedziało, że aktualnie rozsyłana do użytkowników wersja „czerwiec'93" jest ostatnim bezpłatnym *upgradem*. Producent Calamusa oczywiście nie ukrywa swych pretensji do firmy Atari, która przez ostatnie dwa lata wiele zrobiła, by umniejszyć znaczenie Atari w DTP... Ale nie oznacza to od razu rezygnacji z rynku Atari, który do niedawna znajdował się w stanie stagnacji. Zresztą ogólna, kryzysowa sytuacja na rynku komputerowym ostatnich lat też wpłynęła na stan DMC, samego Atari i innych firm z branży.

Dietmar Meyfeldt, szef DMC, jako jeden z powodów zainteresowania się systemem Windows NT wymienił potrzebę powiększenia rynku dla Calamusa, zarówno pod względem liczbowym — ilości zainstalowanych kopii — ale także pod względem mocy sprzętu. Jeśli ktoś myśli, że Windows NT jest systemem przeznaczonym dla zwykłych pecetów, nawet tych 486, to jest w błędzie, przynajmniej przez najbliższy rok, dwa. Jest to system przeznaczony dla szybkich stacji roboczych, serwerów sieci komputerowych, czyli nieco poważniejszych zastosowań niż *personal computing*. Świadczą o tym wysokie wymagania sprzętowe Windows NT oraz wszystkie zawarte w nich zabezpieczenia i mechanizmy sieciowe. Możliwość pracy z Calamusem na stacji roboczej RISC czy szybkim 486 stwarza nowe możliwości użytkownikowi DTP. Calamus NT nie jest przeznaczony dla dotychczasowych użytkowników Ventury czy Page Makera na 286 czy 386, ale dla tych profesjonalistów, którym szybkość „tetetki" czy peceta już nie wystarcza. Potencjalnymi nabywcami tej wersji Calamusa są więc duże studia DTP, posiadacze naświetlarek świadczący usługi itp. Wywodzić się oni będą z dotychczasowych użytkowników Atari a także z innych platform. Część z nich prawdopodobnie będzie równolegle używać zarówno SL na Atari jak i NT na szybszej stacji roboczej, w zależności od zapotrzebowania na moc obliczeniową. Właśnie dlatego DMC zapewniło absolutną zgodność formatu dokumentów Calamusa NT z dotychczasową wersją SL. Podobnie z formatem czcionek, wersja NT posługuje się znanymi nam zbiorami CFN, ale potrafi zrobić użytek także z czcionek postscriptowych. W takim wieloplatformowym studiu usług DTP będzie sobie stało kilka „tetetek" jako stanowiska do składu i edycji, a stacja robocza będzie służyła jako serwer naświetlarki, ewentualnie do obróbki bardzo obszernych (a przez to wymagających szybkiego komputera) grafik rastrowych i prac retuszerskich, za pomocą innych programów.

Pojawienie się Calamusa w wersji dla NT poszerza więc zakres zastosowań tego programu o jeszcze bardziej profesjonalną część rynku, dla której dotychczasowe Atari byłoby zbyt słabym komputerem. Ugruntowuje to pozycję Calamusa a przez to także pozycję Atari jako komputera do profesjonalnego DTP. Kogo stać na stację graficzną za 20 tysięcy marek kupi sobie NT, do tego ze dwa, trzy komputery Atari TT do mniejszych prac — i szafa gra. Kto zaś nie potrzebuje tak wielkich mocy „przerobowych" lub go po prostu nie stać, a podoba mu się idea Calamusa — kupuje Atari. Szafa też gra!

Przyszłe szybsze komputery naszej ukochanej firmy, jak tylko się ich doczekamy, stworzą jeszcze jeden szczebel pośredni pomiędzy TT a stacjami RISC. Pewne nadzieje można też wiązać z mającymi się pojawić szybszymi klonami TT, produkowanymi przez firmę GE-Soft (patrz „Newsy"). Szerokie spektrum maszyn dla Calamusa — na różne kieszenie i potrzeby.

### NT — take a closer look.

W niemieckim czasopiśmie „ST-Magazin" znajduje co miesiąc swe miejsce dodatek poświęcony DTP na Atari — magazyn „Cicero". W numerze 6/93 autorzy tego dodatku opisali stację roboczą M700-10 firmy Olivetti, na której pracuje nowy Calamus NT, pod kontrolą systemu operacyjnego Windows NT. Dysponowali oni wersjami beta zarówno Calamusa jak i systemu NT. Jest to stacja typu RISC, wyposażona w nowoczesny procesor MIPS Technology R4000PC, taktowany zegarem 50 MHz, o pamięci RAM 32 MB (model podstawowy). Do tego jest pół giga dysku twardego, CD-ROM i karta sieciowa. Monitor 1280 x 1024 w 256 kolorach o przekątnej 20 cali. Wszystko to za cenę około 25 tysięcy marek, razem z systemem operacyjnym i Calamusem. W Polsce dojdzie cło i podatek, razem około 30–35 tys. marek.

Firma DMC zawarła umowę z Olivetti a także z firmą Agfa. Wszystko to ma na celu zapewnienie stałych, „cieplarnianych" warunków dla rozwoju Calamusa. Właśnie ze względu na te umowy, na targach CEBIT'93 Calamus NT był prezentowany na stanowisku firmy Olivetti, dlatego pierwsza pojawiła się wersja dla procesora MIPS R400PC, w który wyposażone jest wspomniane Olivetti. W przyszłości dopiero przewidywana jest wersja dla procesorów Intel (dopiero wtedy będzie można mówić o Calamusie na pececie 486), oraz wersja na procesor Alpha.

Komentarze „Cicero" odnośnie nowej platformy są chyba prawidłowe: przez „okna NT" wlatuje świeży wiatr na rynek Atari-DTP, czy jak kto woli Calamus-DTP. Nowa platforma otwiera nowe możliwości i zwiększa moce przerobowe, co oznacza poszerzenie rynku o NT, a nie przesiadkę na NT. No jasne, fani Atari mogą odczuwać niejaką zazdrość o nową oblubienicę firmy DMC, ale to tylko pierwszy szok, po nim przychodzi duma z tak wielce obiecującego dziecka, jakim jest Calamus.

### A co na starym podwórku?

Póki co, DMC utrzymuje się z rynku Atari i nie zapomina o tym. Wprowadzona na rynek nowa wersja SL zawiera kilka ważnych nowości, które pokrótce scharakteryzujemy.

Kotwiczki. Pojęciem tym oznacza się „przyczepienie" ramki (zwykle grafiki rastrowej lub wektorowej) do ustalonego miejsca w ramie tekstowej. Do czego to służy? Wyobraźmy sobie, że w tworzonym dokumencie używamy (w tekście) jakiejś nazwy, którą zwykle pisze się tzw. fontem firmowym. Niech to będzie napis „Coca Cola". Dysponujemy takim firmowym napisem, jako grafiką wektorową. Chcielibyśmy teraz, aby wszędzie, gdzie w tekście jest napisane „Coca Cola" występowała taka właśnie ładna grafika. Bez użycia kotwiczek należałoby skopiować odpowiednią ilość razy ramkę z grafiką i poustawiać ręcznie wszystkie kopie w odpowiednich miejscach tekstu. W przypadku przeformatowania tekstu, zmiany wielkości litery itp. cała robota z ustawianiem „Coca Coli" musi być wykonana od początku. Gdy użyjemy kotwiczek wystarczy raz przypisać ramkę wektorową do odpowiednich miejsc w tekście, a ramki będą wędrować same za przeformatowywanym tekstem. Idea kotwiczek w Calamusie może być zastosowana do dowolnego typu ram, nie tylko grafiki, dlatego można pomyśleć o kilku innych zastosowaniach, w tabelach, spisach adresów z nazwami firm itp.

Drugą nowością są tabulatory wypełniające. Używane są przykładowo w spisach treści, gdzie po tekście oznaczającym np. nazwę rozdziału następują kropki aż pod prawy margines strony, gdzie znajduje się liczba oznaczająca stronę na której znajduje się początek rzeczonego rozdziału. Te kropki to właśnie tabulator wypełniający. Nie potrzeba martwić się o „wstukanie" odpowiedniej ilości kropek, Calamus zrobi to za nas, aby wypełnić pustą przestrzeń o zadanych rozmiarach.

Bardzo przydatną cechą nowego SL-a jest możliwość obracania grafiki rastrowej o dowolny kąt. Brak tej możliwości był często krytykowany przez dotychczasowych użytkowników. Dla funkcji tej powołano cały nowy moduł, pozwalający obracać ramki grafiki monochromatycznej, 8-mio, 24-ro i 32-bitowej. Dokładność wynosi setne części stopnia. Przy obracaniu można dodatkowo włączyć antialiasing zmniejszający straty jakości wynikające z obracania rastra.

Kolejnym nowym modułem Calamusa jest LineArt. Zastępuje on praktycznie dotychczasowy moduł wektorowy i oferuje niektóre funkcje programu Outline Art — dodatkowego narzędzia wspomagającego dotychczas Calamusa z zewnątrz. Oprócz manipulacji krzywymi wektorowymi, tekstem na krzywej mamy do dyspozycji generator przejść pomiędzy obiektami wektorowymi. Nie jest to co prawda *morphing*, gdyż obiekty muszą być do siebie w miarę podobne, ale i tak Calamus zyskał nowe możliwości. Generowane przy tym przejścia barwne mogą być zamieniane na grafiki rastrowe.

*Rastercache* to nowe pojęcie, którego użytkownicy Calamusa muszą się nauczyć. Technologia softrippingu, którą wylansowała firma DMC przewiduje między innymi dużą elastyczność przy dobieraniu rastra naświetlanego dokumentu. Aby osiągnąć najwyższą jakość reprodukcji i ominąć niebezpieczeństwo pojawienia się tzw. mory (nieprzyjemne „interferencje" na skanowanym zdjęciu) raster generuje się nieraz osobno dla czterech wyciągów barwnych, osobno dla każdej strony lub nawet ramki. Dwie godziny oczekiwania na dobry raster nie było dotychczas rzadkością. Co prawda efekty pozwalają być dumnym z Calamusa, ale dla profesjonalistów czas to pieniądz.

Nowy generator rastra posługuje się technologią *rastercacheing*. W pliku dyskowym można przechowywać wygenerowane rastry dla poszczególnych „sytuacji" edytorskich, do ponownego wykorzystania przy następnych naświetleniach. Tym użytkownikom, którzy nie posiadają naświetlarki, lecz korzystają z usług firm trzecich pozwala to na wygenerowanie samemu odpowiednich rastrów samodzielnie i dostarczenie do naświetlenia dokumentu wraz z odpowiednim plikiem wygenerowanych wzorców rastra, co przyspieszy naświetlanie i zminimalizuje czas trwania usługi, a tym samym zmniejszy cenę.

Dotychczasowi użytkownicy Calamusa denerwowali się, że nie bardzo można zautomatyzować wstawianie tzw. paserów do kolorowego dokumentu. Kolorowa technika drukarska opiera się na czterokrotnym zadrukowywaniu powierzchni papieru podstawowymi farbami (CMYK), co przez zmieszanie daje możliwość osiągnięcia prawie każdej barwy. Konieczność czterokrotnego przejścia arkusza papieru przez maszynę powoduje możliwość przesunięcia się poszczególnych barw podstawowych o ułamki milimetra, co daje nieprzyjemne efekty i nieostre kontury. Aby tego uniknąć na marginesach drukowanej pracy umieszcza się pasery, czyli kolorowe paski i linie synchronizujące, pozwalające na stałą kontrolę zbieżności barw. Owe paski i linie trzeba było dotychczas „generować" ręcznie. Co prawda wprawny użytkownik zautomatyzował sobie ten proces na ile to było możliwe (użycie clipboardu-schowka czyli calamusowego magazynu). Obecnie program w pełni automatycznie potrafi wprowadzić do dokumentu pasery, linie cięcia, nazwy barw podstawowych CMYK oraz punkty łączenia matryc, jeśli dokument składa się z większej ich ilości (np. dokumenty dwustronne).

Na deser DMC proponuje nam jeszcze kilka innych drobiazgów takich jak system pomocy (!) oraz moduł o tajemniczej nazwie Farbplane Swapper.

**Calamus S — akt drugi.**

Jak gdyby z rozpędu programiści z Walluf (siedziba firmy) pokazali światu nowego Calamusa S — wersję 2.0. Pracuje ona poprawnie w nowych rozdzielczościach Falcona, przez co czyni z tego komputera nową atrakcyjną cenowo platformę dla popularnych zastosowań DTP. Do tego Calamus S 2.0 posiada niektóre nowe moduły z opisanej wyżej najnowszej wersji SL. Konkretnie chodzi tu o moduł obracania grafik rastrowych, Farbplane Swapper i system pomocy.

Calamus S pracujący w 256 kolorach na Falconie jest atrakcyjnym stanowiskiem *middle class*, czyli dla początkujących, ale poważniejszych użytkowników, których nie stać na zainwestowanie w TT z kartą graficzną. Obecnie Falcon030, mimo domowego charakteru, jest maszyną o potencjalnie dużych możliwościach rozbudowy. Włożony w obudowę tower, wyposażony w „dopalacz" 32 MHz i większą ilość pamięci Fast-RAM już dziś osiąga moc obliczeniową TT przy lepszej grafice (bez użycia dodatkowych kart graficznych). Do tego mamy potężny procesor DSP, który dziś stosowany jest głównie w zastosowaniach muzycznych, ale w przyszłości można spodziewać się dalszego zwiększenia ilości oprogramowania graficznego wykorzystującego procesor sygnałowy. Najprostszym zastosowaniem jest szybka kompresja i dekompresja obrazów oraz operacje na grafice, filtrowanie, transformacje itp. — przeliczane za pomocą DSP.

**Kasa.**

Na koniec wypada podać ceny wszystkich tych przyjemności, które opisane są w tym artykule. Nowa wersja SL, jak już wspomniano, jest bezpłatna dla dotychczasowych użytkowników. Przesiadka na wersję NT kosztuje 989 DM i przewidywana jest na wrzesień'93. Kto wcześniej zakupił za tą samą kwotę zestaw 99 czcionek DMC-Classic-Types ten kupon na Calamusa NT dostał bezpłatnie.

Właściele Calamusa S za wersję 2.0 swojego programu muszą zapłacić 149 DM. Za 198 DM mogą dodatkowo otrzymać moduł Paint&Draw znany z przedostatniej „poprawki" SL. Normalnie kosztuje on 398 DM. Przesiadka na Calamusa NT kosztować będzie bagatela — 1679 DM. Sprawdza się w tym przypadku stara prawda, że oprogramowanie na peceta jest w większości wypadków droższe od funkcjonalnych odpowiedników w systemie Atari...



***Rafał Komorowski***

# ST^E fan w Holandii

Jakiś czas temu przyszło do redakcji zamówienie na prenumeratę „ST^Efana", którego nadawca mieszka w... Holandii. Ponieważ jego nazwisko — Rein Bakhuizen van den Brink — także nie zabrzmiało swojsko, było tylko jedno wytłumaczenie: prenumeratę „ST^Efana" chce uzyskać Holender!!! Zapewne domyślacie się, jak podniosły i dumny nastrój wywołało to w redakcji. Nasuwały się nam pytania: dlaczego Holender? Czy zna język polski? Może to nauczyciel lub lingwista? Pierwszy numer czasopisma oczywiście wysłaliśmy, razem z listem zawierającym nasze pytania. Prenumeratę Panu van den Brink postanowiliśmy ufundować za darmo — przesłanie należności do Polski sprawiłoby na pewno dużo kłopotu i kosztów, poza tym taki zaszczyt...

Po miesiącu czy dwóch otrzymaliśmy odpowiedź (w języku polskim!), która wyjaśniła nam wszystko. Stopień opanowania naszego języka wzbudził nasz podziw, ale mimo to „ugładziliśmy" jego tekst dla potrzeb „ST^Efana". Oto list od naszego Czytelnika z Holandii:

*Szanowna Redakcjo!*

*Bardzo dziękuję za list i przyjemną niespodziankę, którą mi zrobiliście.* (Chodzi o prenumeratę — przyp. red.)

*Skończyłem studia matematyczne i zająłem się informatyką, ponieważ wtedy, w latach siedemdziesiątych, nie było u nas pracy dla matematyków. Pracowałem jako programista w biurze, przy komputerach PDP 11/34 i VAX. Kiedy zdecydowałem się, że trzeba kupić komputer do domu to zaistniał problem, bo nie chciałem żadnych IBM/MS-DOS-ów. Na szczęście fotograf od nas z pracy miał dobrą alternatywę w domu: Atari Mega ST 2. I tak się stało. Od początku 1989 roku zacząłem pracować z Atari i do dziś mam ten sam komputer — w międzyczasie pojawiło się 4 MB pamięci, twardy dysk 210 MB oraz TOS 2.06.*

*W biurze używamy obowiązkowo Wordperfect 5.0/5.1 i ja też znam już dobrze ten program, ale nie podoba mi się on za bardzo. W domu wystarczą mi That's Write 2.05, Calamus S 2.0, Type Art i C-Font.*

*Pierwszy raz przyjechałem do Polski w 1974 roku, zakochałem się w Polsce i w Polce. Trzy lata później braliśmy ślub w Krakowie. Mieszkamy już dawno w Holandii, żona dobrze mówi po holendersku. Nasz syn Lech — pięć i pół roku — najpierw mówił po polsku (tzn. w domu cały czas mówiliśmy po polsku) i dopiero wiele potem (w szkole i u babci) zaczął mówić po niderlandzku.*

*List ten został napisany w programie That's Write z fontami pochodzącymi z Calamusa (Public Domain). Type Artem — to dobry edytor fontów do Calamusa — zrobiłem wszystkie „polskie ogonki". Trudno mi było się zgadzać ze standardem (którym? ISO-LATIN 2, DHN, Mazovia??). Skoro ja nie chcę zrezygnować z francuskich, hiszpańskich i niemieckich akcentów i umlautów, to zmieniłem kilka znaków hebrajskich na polskie. ASCII 194–202 to „ćłńśźżąęó" natomiast 203–211 to „ĆŁŃŚŹŻĄĘÓ".* (Co za wspaniały zbieg okoliczności! Przecież Pan odkrył niemalże po raz drugi standard Klubu Atari! Rozbieżności są w zasadzie niewielkie — przyp. red.) *Na klawiaturze mam je na „numeric keypad". That's Write ma możliwość ustalania klawiatury przez użytkownika. Po napisaniu tekstu może on być wprowadzony jako ASCII do Calamusa. Czcionki z Calamusa (te z ogonkami) mogę przekonwertować za pomocą programu C-Font na fonty TWrite'a lub GEM-u.*

*Grupa wielbicieli Atari, jak na całym świecie, w Holandii jest pod wielką presją. Jest coraz mniej sklepów, gdzie można dostać sprzęt albo programy. Na początku tego roku mieliśmy w Holandii trzy czasopisma: ACN Nieuws, ST oraz Start. Start niedawno splajtował.* (Oprócz tego splajtował także holenderski oddział Commodore, więc nie ma co się smucić... — przyp. red.)*Podobno gorzej w Niemczech: ja miałem w styczniu jeszcze Atari Journal, ST Computer, ST Magazin oraz TOS Magazin. Atari Journal już w lutym nie było, ostatni TOS w lipcu i ostatni ST Magazin w sierpniu! Tylko ST Computer może jeszcze wytrzymać.* (Tak, to prawda, obecnie ukazuje się jedynie ST Computer, który nosi także na okładce znaczki tych dwóch innych gazet: TOS i ST Magazin. Połączyły one siły i ukazują się razem pod jednym tytułem: ST Computer — przyp. red.) *A w Polsce?? Świat Atari, Tajemnice Atari, Atari Magazyn, ST-fan i ST^Efan. Czy jest naprawdę rynek dla wszystkich? Czy nie lepiej współpracować?*

*W Holandii Atari było/jest bardzo popularne wśród studentów, nieraz dlatego, że z dodatkowym Spectre GCR był/jest najtańszym Macintoshem! Mam rzeczywiście Spectre jak i Vortex Atonce 386, ale są bardzo rzadko używane.*

*Ile jest posiadaczy Atari (ST, TT, Falcon) w Holandii nie wiem. Może około 20.000. Długo musieliśmy czekać na Falcona, ale wreszcie można go kupić. Ja na razie go nie potrzebuję, wolę poczekać na Falcona 040 — czy w ogóle będzie? Programy pochodzą w większości z Niemiec. Ceny są podobne, może trochę wyższe. Rok temu DMC sprzedawał swoje produkty bardzo tanio, chyba taniej niż w Niemczech. Kupiłem wtedy Calamusa S za 300 florenów (265 DM) i Type Art za f425. Moduły Calamusa kosztują od f50 do 100, niektóre 200. Jako edytor tekstowy That's Write jest bardzo popularny. 1ST Word Plus nigdy nie używałem* (Nie ma czego żałować — R.K., M.K.) *Dobrym programem okazał się Papyrus. Holenderską wersję zrobił sklep w Leiden (to tylko pięć minut drogi na rowerze) i co tydzień można było zobaczyć postępy. (...) Jak już pisałem są jeszcze dwa czasopisma holenderskie. Wydawnictwa powstałe z grup użytkowników Atari. ST to grupa z Leiden, raczej studencka i mało komercyjna, natomiast ACN w Haarlemie bardzo rozbudowało się i przekształciło w dom wysyłkowy. Jak niedawno splajtował importer Signum, to oni, ACN, już teraz sprowadzają produkty Application Systems Heidelberg. Wiosną czekałem miesiąc na update Signum 3.3 dzwoniłem do importera i słychać było dziwny dźwięk, że wyłączyli telefon. Trochę później dowiedziałem się, że tej firmy już nie ma. Dwa tygodnie temu jechałem do Haarlemu i odebrałem od ACN najnowszą wersję Signum. Nie jest jeszcze gotowy ich „final" albo "Ultimate" Virus Killer wersja 6.0. Ten program napisany przez Richarda Karsmakersa jest bardzo znany również w Anglii i w Polsce. ACN też ma największy wybór fontów PD do Calamusa. Dołączę ich czasopismo, wydaje mi się, że warto zobaczyć pomimo, że jest po holendersku. Jeśli macie jeszcze jakieś konkretne pytania to proszę, teraz muszę się śpieszyć, bo mam o godz. 17.00 autobus do Polski. Ten list wyślę z Krakowa.*

*Pozdrowienia*

*Rein Bakhuizen van den Brink.*

Dołączona do listu holenderska gazeta „Atari ST Nieuws" wydawana przez wspomniany ACN (Atari ST Computerclub Nederland) zrobiła na nas duże wrażenie. 112 stron formatu A3. Z listu wynika, że Polska ze swoimi czasopismami jest w światowej awangardzie Atari! Sądzimy, że sytuacja w krajach Beneluksu jest nieco lepsza, niż sugerowałby to list naszego holenderskiego przyjaciela. Tym bardziej, że obecnie w Holandii znajduje się europejska centrala Atari i magazyn, z którego zaopatrują się oddziały w Niemczech, Francji, Wielkiej Brytanii...

## Komputery ATARI

| | |
|---|---|
| 1. 520 STFM | 5,100 |
| **2. 1040 STFM** | **6,000** |
| 3. 1040 STE | 7,400 |
| 4. 1040 STE 2 MB RAM | 8,900 |
| 5. FALCON030 1 MB RAM | 21,500 |
| 6. FALCON030 4 MB RAM | 25,300 |
| 7. 4TT030 | 42,600 |
| 8. SM 144/146 | 3,600 |
| 9. TTM 194/195 | 21,300 |
| **10. SC 1224** | **4,500** |
| 11. SC 1435 | 6,900 |
| **12. SLM 605** | **16,500** |
| **13. SLM 804** + 600 dpi | **23,400** |
| 14. TONER SLM 605 TWIN PACK | 1,500 |
| 15. TONER SLM 804 | 1,[illegible]00 |
| 16. BĘBEN SLM 605 | 2,800 |
| 17. BĘBEN SLM 804 | 4,700 |

## TT I DTP...

| | |
|---|---|
| 1. TOWER | 8,000 |
| 2. ATARI 16 MB EXP. PUSTY | 3,600 |
| 3. MIGHTY MIC 32 PUSTY | 7,200 |
| 4. MIGHTY MIC 64 PUSTY | 14,400 |
| 5. SIMM 4 MB (60 ns) | 6,000 |
| 6. QUANTUM 85 MB | 4,900 |
| 7. QUANTUM 120 MB | 6,500 |
| 8. QUANTUM 240 MB | 10,700 |
| 9. SYQUEST 44 MB | 9,600 |
| 10. CARTRIDGE 44 MB | 2,300 |
| 11. MAGIC 256 KOLORÓW | 6,000 |
| 12. CALAMUS 1.09 N | 1,600 |
| 13. INKAUST | 2,000 |
| 14. 1 ST WORD PLUS | 0,700 |
| 15. CALAMUS SL | 17,500 |
| 16. OUTLINE ART 3.0 | 10,000 |
| **17. DA'S LAYOUT (DIDOT + RETOUCHE + VECTOR)** | **12,[illegible]00** |
| 18. PHASE IV | 11,600 |

## HP & EPSON

| | |
|---|---|
| 1. HP IV | 51,700 |
| 2. HP IV L | 23,900 |
| 3. HP 510 | 10,000 |
| **4. HP 500C** | **13,000** |
| 5. HP 550C | 23,500 |
| 6. EPSON GT 6500 CENTRONICS | TEL. |
| 7. EPSON GT 8000 SCSI | TEL. |
| 8. DRIVERY DRUKAREK HP DLA CALAMUSA | 0,800 |
| 9. GT LOOK | 4,000 |

## [illegible]

| | |
|---|---|
| 1. DESKTOP + K[illegible] | 6,200 |
| 2. MIGHTY SONI[illegible] | 11,500 |
| 3. CONNER 85 MB WEWN. | 5,600 |
| 4. OBUDOWA DYSKU TW. + KABEL SCSI II | 5,600 |
| **5. SCREENBLASTER** | **2,200** |
| 6. MATRIX DIGITIZER | 14,200 |
| 7. FALCON SPEED | 7,300 |
| **8. OVERLAY** | **2,900** |
| **9. PAPYRUS 2.0** | **2,100** |
| 10. XENOMORPH 3D 2.0 | 2,500 |
| 11. DA'S VECTOR | 3,600 |
| 12. CHAGALL COLOR | 10,000 |
| 13. DIGITAPE LIGHT | 3,300 |
| 14. MUSICOM 1.0 | 1,500 |
| 15. NVDI 2.5 FALCON | 0,700 |
| 16. MONO SVGA 14" | 3,000 |
| 17. CRYSTAL 14 S SVGA KOLOR 14" | 7.800 |

## EMAGIC

| | |
|---|---|
| 1. NOTATOR LOGIC | 13,200 |
| 2. NOTATOR SL | 8,200 |
| 3. NOTATOR ALPHA | 5,000 |
| 4. CREATOR SL | 5,500 |
| 5. HEARMASTER | 2,800 |
| 6. SOUNDSURFER | 3,400 |
| 7. UNITOR 2 | 7,600 |
| 8. EXPORT | 2,600 |

## ST I EDUKACJA

| | |
|---|---|
| 1. POKRYWA NA KLAW. | 0,120 |
| 2. MOUSE BRUSH | 0,600 |
| **3. KAWAI PSS 470 + HAPPY MUSIC** | **2,200** |
| 4. STACJA DYSKÓW 3,5" | 1,800 |
| 5. STACJA DYSKÓW 5,25" | 2,200 |
| **6. VORTEX ATONCE+** | **3,500** |
| **7. SCSI KONTROLER** | **1,200** |
| 8. KABEL ACSI D13-D19 | 0,150 |
| 9. DISCOVERY PACK | 0,320 |
| 10. ST ORTOGRAFIA | 0,060 |
| 11. ST TRIVIA | 0,090 |
| 12. ST SŁOWNIK NIEMIECKI | 0,090 |
| 13. ST SŁOWNIK ANGIELSKI | 0,090 |
| 14. SAPER | 0,100 |
| 15. TOTO LOTEK | 0,100 |
| 16. MAT PUZZLE | 0,100 |
| 17. WIESŁAW (WIDEO WYPOŻYCZALNIA) | 0,800 |
| 18. KOSZTORYSY | 1,000 |

Wszystkie ceny dotyczą sprzedaży sklepowej i są podane w milionach zł. Ceny nie obejmują kosztów wysyłki, natomiast zawierają podatek VAT. Wysyłka poprzez firmę SERVISCO z dostawą do domu odbiorcy. Realizacja zamówienia trwa minimum 14 dni. Przy sprzedaży wysyłkowej proszę korzystać z dołączonej karty zamówieniowej. Wpłaty prosimy kierować na konto: **BPH IIo/Wrocław 329215-69368-136.** Proszę czytelnie wypełniać dowody wpłaty, umieścić dokładne dane personalne, adres, telefon, datę i przedmiot zamówienia. W przypadku większych zamówień (ponad 1 milion zł) konieczne jest dołączenie do karty zamówieniowej kopii dowodu wpłaty. W przypadku zamówień do 1 miliona zł istnieje możliwość wysyłki pocztowej za zaliczeniem. Wszystkie urządzenia posiadają roczną gwarancję. Duże systemy DTP i Multimedia podlegają opcjonalnie 48-godzinnemu serwisowi.

## ATARI Lynx

| | |
|---|---|
| 1. LYNX II | 2,890 |
| 2. ZASILACZ | 0,300 |
| 3. ZASILACZ Z ZAPALNICZKI SAMOCHOD. | 0,370 |
| 4. COM LYNX | 0,190 |
| 5. TORBA DO LYNXA | 0,370 |
| 6. SUN VISOR | 0,150 |

GRY DO LYNXA:

| | |
|---|---|
| 7. Batman Returns, Dracula Undead | po 1,100 |
| 8. A.P.B., Awesome Golf, Basketbrawl, Battlezone, Bil & Teds Excellent Adventure, Blood/Guts Hockey, Blue Lighting, California Games, Checkered Flag, Crystal Mines, Dirty Larry, Hard Drivin', Heavyweight Contender, Hockey, Hydra, Kung Food, Lynx Casino, Malibu beach Vollyball, NFL Football, Ninja Gaiden, Pinbal Jam, Power Factor, Rann-Den, Roadblasters, Scrapyard Dog, Shadow of the Beast, Steel Talons, STUN Runner, Supersqueek, Switchblade 2, Toki, Tournament Cyberball, Turbo Sub, Viking Child, Warbirds, Xybots | po 0,900 |
| 9. Baseball Heroes, Rampage, Rampart, Rygar | po 0,800 |
| 10. Chips Challange, Ms Pacman, Pacland, Paperboy, Shanghai, Tedds Adventure in Slime World, Vindicators, World Class Soccer, Xenophobe | po 0,650 |
| 11. Blockout, Electrocop, Gates of Zendecon, Gauntlet the Third Encounter, Klax, Robosquash, Robotron, Zalor Mercenary | po 0,480 |

## LITERATURA

| | |
|---|---|
| 1. CALAMUS 1.09 N | 0,100 |
| 2. CALAMUS SL | 0,270 |
| 3. OUTLINE ART | 0,100 |
| 4. BIBLIOTEKA GRAFIK | 0,120 |
| 5. KATALOG FONTÓW DO CALAMUSA | 0,120 |
| 6. SCIGRAPH | 0,120 |
| 7. ADIMENS ST | 0,100 |
| 8. DB MAN V. 4.0 DLA ST | 0,070 |
| 9. INTERN ST | 0,120 |
| 10. GEM ST | 0,120 |
| 11. PROFIBUCH ST, TT | 1,000 |




**Atar [illegible]ystem**
**ul.Tr[illegible]ska 12, [illegible]412**
**53-67[illegible] [illegible]**
**tel/fax ([illegible]) [illegible]-64-60**

ATARI®

Dariusz Paździor

# Persistance of Vision Ray

**Na pewno wielu z Was oglądało w telewizji modne ostatnio reklamy, czołówki programów stworzone przy pomocy komputerów. Są one przygotowywane na maszynach o dużej mocy obliczeniowej (często są to systemy oparte na procesorach typu RISC) przy pomocy specjalistycznego oprogramowania graficznego. Ceny takich zestawów znacznie (delikatnie powiedziane...) przekraczają możliwości przeciętnego „naciskacza klawiszy". Niewielu jednak wie, że równie dobre efekty można uzyskać używając domowych lub półprofesjonalnych komputerów Atari oraz dostępnego na polskim rynku oprogramowania komercyjnego oraz shareware. Do pierwszej grupy zaliczyć można między innymi programy Xenomorph 3D (ostatnio wersja 2.0), Chronos 3D oraz częściowo CAD 3D, zaś do drugiej — QRT Ray Tracer i PoV-Ray.**

Właśnie temu ostatniemu programowi poświęcony jest cały cykl artykułów. Autorami PoV'a są: David Buck i Aron Collins. Jest on oparty na programie DKB Ray Tracer 2.12 napisanym na Amigę przez tych samych autorów. Dla zainteresowanych podaję adres faceta, który jest szefem POV Ray Team:

Drew Wells
CIS: 73767,1244
AOL: Drew Wells
Prodigy: SXNX74A
US Mail:
905 North Ave 66
Los Angeles, CA
90042
Tel.ó: (213) 254-4041

Nowych wersji programu szukajcie bądź w najbliższym (zagranicznym BBS), bądź w redakcji ST'fana.

Cykl artykułów obejmuje podstawy teoretyczne grafiki trójwymiarowej, modelowania oświetlenia oraz dokładny opis języka jakim definiuje się wszelkiego rodzaju obiekty w przestrzeni. Dodatkowo na kolejnych kilku dyskietkach ST'fana znajdziecie kompletny pakiet oprogramowania POVRAY wraz z dokumentacją.

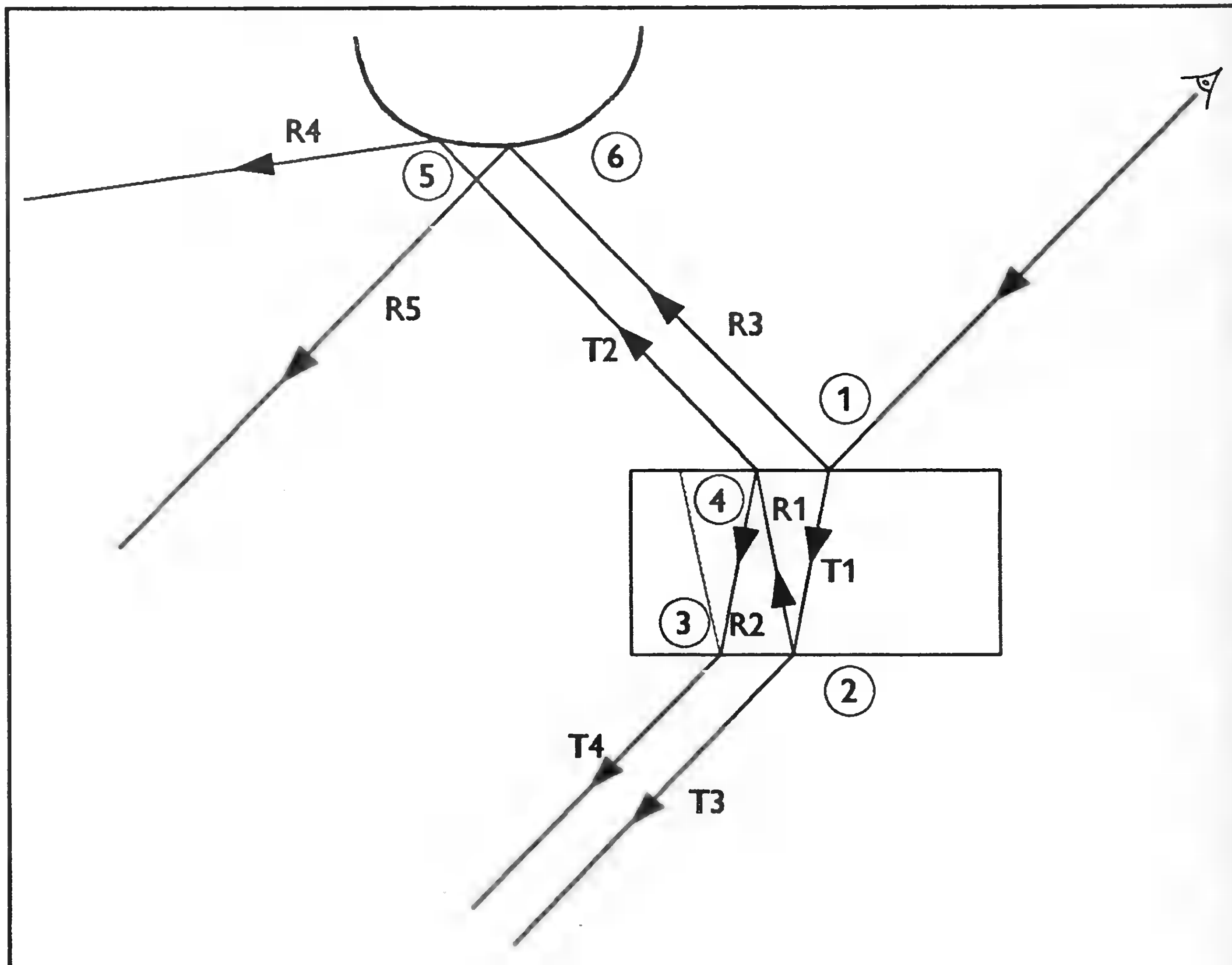
## Śledzenie promieni — co to jest ?

Od wielu lat informatycy starają się opracować najlepszy i najbardziej efektywny algorytm wyznaczania cieni i linii zasłoniętych, modelowania oświetlenia i tekstury. W wyniku tych prac powstały bardzo ciekawe algorytmy realizujace te zadania miedzy innymi: „dziel i zwyciężaj", polegający na podziale obszaru ekranu na coraz mniejsze fragmenty zawierające tylko obiekt, algorytm z buforem głębokości, będący jednym najłatwiejszy do implementacji. Metoda ta polega na wypełnianiu ekranu znajdując dla każdego punktu ścianę leżącą najbliżej na prostej łączącej punkt obserwacji z danym punktem i przypisując mu kolor/jasność odpowiedniego fragmentu ściany. Powstało także wiele, wiele innych bardziej lub mniej skomplikowanych, tych którzy pragną się z nimi zapoznać zapraszam do przeczytania pozycji 1 z podanej literatury. Jednak okazuje się, że żaden z tych algorytmów nie daje tak dobrych rezultatów jak metoda śledzenia promieni *(ray tracing)* światła. Algorytm ten jest uproszczoną realizacją fizycznych zjawisk zachodzących w przestrzeni (odbicie, załamanie, rozproszenie światła itp.). Źródła światła emitują nieskończenie wiele promieni w różnych kierunkach. Na ekranie monitora rastrowego widoczny jest jedynie wycinek przestrzeni, zawierający się wewnątrz ostrosłupa widzenia. Oprócz tego generowany obrazu ma ograniczoną i skończoną rozdzielczość. Oznacza to, że do oka obserwatora dociera skończona ilość promieni. W algorytmie śledzenia promieni postępuje się zatem odwrotnie jak w optyce geometrycznej. Od obserwatora (środka rzutowania) prowadzi się promień przez środek każdego punktu w kierunku źródeł światła i śledzi jego bieg. Jeżeli promień napotyka jakiś przedmiot, to częściowo odbija się od niego, a częściowo go przenika. Dalej śledzimy drogę dwóch promieni — odbitego i przepuszczonego, możliwa jest także sytuacja gdy pozostaje do śledzenia tylko jeden promień, gdy ciało jest nieprzeźroczyste (brak promienia przepuszczonego). Kierunek promienia odbitego określamy z równości kąta padania i odbicia, a przy promieniu przenikającym obiekt, uwzględniamy zmiany kierunku na granicy dwóch ośrodków o różnych współczynnikach załamania. Można także brać pod uwagę zjawisko dyspersji (rozszczepienia światła), tj. zależność współczynnika załamania od długości fali świetlnej, czyli jej barwy. W tym przypadku trzeba brać pod uwagę nie składowe RGB ale jego widmo. Każdy z tych dwóch promieni, odbity i przepuszczony, może się dalej odbijać i/lub przenikać inne przedmioty itd.

Drogę promienia od obserwatora przez środek danego piksela w kierunku źródeł światła opisuje się drzewem binarnym. Przy okazji warto wyjaśnić co to jest struktura drzewiasta. Drzewo binarne jest to uporządkowany zbiór elementów zwanych węzłami, które oprócz właściwej informacji zawierają dane o elementach podporządkowanych. W ten sposób węzły nadrzędne i podrzędne są ze sobą powiązane. Gdyby graficznie wykreślić zależności między poszczególnymi elementami to utworzą one obraz drzewa. Jego węzły odpowiadają przecięciom promieni z powierzchniami przedmiotów, a gałęzie promieniom odbitym i przepuszczonym. Liśćmi (węzłami termalnymi) drzewa są punkty powierzchni oświetlane bezpośrednio przez źródła światła. W tych punktach obliczamy natężenie światła wykorzystując fakt, że siła oświetlenia maleje proporcjonalnie do kwadratu odległości od źrodła światła. Następnie, dla wyznaczenia oświetlenia odpowiadającego danemu pikslowi przeglądamy drzewo od liści do korzenia. W każdym węźle uwzględniamy współczynnik odbicia światła od powierzchni i charakterystykę przenikania danego materiału. Zasadniczy koszt algorytmu (nawet do 95 % wykonywanych działań) związanych jest z obliczaniem punktów przecięć promieni z powierzchniami obiektów. Można znacznie uprościć i przyspieszyć obliczenia określając najpierw kule opisującą cały dany przedmiot. W tym przypadku wstępne określenie czy promień go przecina jest bardzo łatwe i szybkie, wystarczy znaleźć odległość środka kuli od promienia (prostej o zadanym równaniu) i porównać z jej promieniem. Jeżeli odległość promienia jest większa od promienia kuli opisującej obiekt to znaczy, że przedmiot i prosta zawierająca promień są rozłączne. Fakt ten jest bardzo ważny gdyż umożliwia znaczne przyspieszenie tworzenia obrazu. Wystarczy podać środek i promień kuli opisującej dany obiekt.

## Persistence of Vision Ray Tracer — pierwszy rzut oka

Ten pierwszy rzut oka wcale nie jest dla programu POV korzystny. Jest to program TTP (cała fura parametrów), wymagający opisania realizowanej sceny za pomocą pliku tekstowego, co wiąże się z conajmniej kilkunastominutowym siedzeniem z kalkulatorem nad kartką i przy edytorze tekstowym — słowem mało zachęcająca perspektywa. Gdzie mu tam do takich programów jak Chronos 3D czy Xenomorph 3d z porządnymi edytorami obiektów 3D. Jednakże po zapoznaniu się z dołączoną dokumentacją i wykonaniu paru przykładowych obrazków (zabijają !!!) mina mi się poprawiła. Wyjaśnił się także powód zastosowania



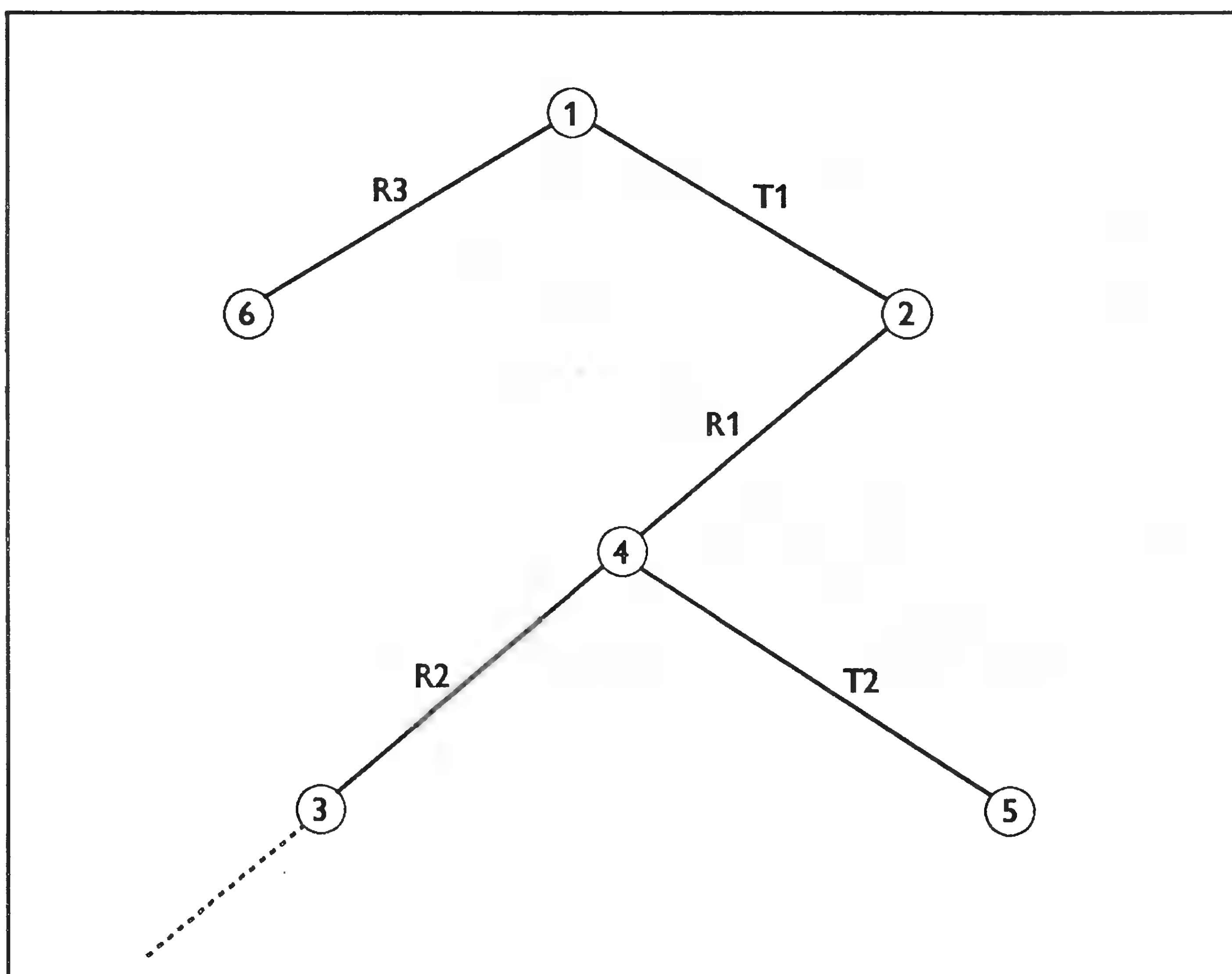
*Idea metody śledzenia promieni*

specjalnego języka oraz programu TTP. Otóż program ten został napisany w jezyku C z myślą rozpowszechnienia go na wszystkich platformach sprzętowych bez względu na możliwości graficzne i obliczeniowe. Dotychczas stworzono wersje POV-Ray na następujące komputery i systemy operacyjne:

— IBM PC oraz kompatybilne z procesorami INTEL 386 lub 486
— Atari ST, TT, Falcon
— Apple Macintosh z procesorem minimum MC68020
— Amiga 500, 1000, 2000, 3000 pożądany AMIGA DOS
— HP RS-6000
— Sun
— Cray
— Vax
— wszystkie komputery pracujące pod systemem UNIX i jego klonami

Jak widać program doczekał się wielu wersji na różnorodne komputery, w tym także na bardzo duże i wydajne maszyny trasputerowe. Jedynym warunkiem na to aby stworzyć wersję POV'a na nowy komputer jest posiadanie kompilatora jezyka C zgodnego z normami ANSI oraz listingu programu POV-Ray.

Opis programu warto zacząć od krótkiej charakterystyki. Oto ona:



*Drzewo opisujące drogę promienia*

— POV-Ray wyposażony jest w łatwy do używania język definiujący obiekty, tekstury oraz oświetlenie
— rozdzielczość generowanego obrazu to maksymalnie 32767 x 32767
— duża biblioteka predefiniowanych obiektów, kolorów oraz tekstur
— zastosowanie tzw. konstruktywnej geometrii brył (CSG — ang. *contructive solid geometry*). Termin ten oznacza metodę budowania brył w wyniku składania ustalonych, elementarnych „klocków". Na takich obiektach można dokonywać analogicznych operacji jak na zbiorach, czyli dodawanie, odejmowanie, część wspólna. Jest to możliwe dzięki zdefiniowaniu wszystkich obiektów jako tzw. obszarów normalnych. Obszar normalny jest to zbiór punktów (x, y, z) spełniających nierówność

$$f(x,y,z) < 0$$

gdzie f jest funkcją na ciągłą na całej przestrzeni $R^3$.

POV-Ray umożliwia definiowanie własnych funkcji (w pewnym zakresie) co pozwala otrzymać najbardziej niezwykłe figury np.: torus, hyperboloid, elipsoid, paraboloid itd.

— przedmioty składające się z wielu elementów można połączyć w jedną całość co umożliwia bardzo łatwe manipulowanie takimi agregatami
— możliwość generowania trójwymiarowych fraktali tzw. height fields
— tworzenie obrazu w trzech formatach Targa (*.TGA), Dump (*.DIS), Raw (*.R8, *.G8, *.B8)
— wiele wbudowanych tekstur np.: Marble, Checkerboard, Wood, Bumpy, Agate, Clouds, Granite, Ripples, Wawes, Leopard, Wrinkled
— użytkownik może utworzyć własne tekstury lub użyć już zdefiniowane w bibliotekach
— możliwość przerwania obliczeń w każdej chwili lub tworzenia obrazka na raty czyli po kawałku

## POV-Ray — parametry wejściowe

Na dysku dołączonym do ST^E^fana znajdują się trzy programy TTP: POV68000.TTP, POV68030.TTP oraz POV68881.TTP. Pierwszy z nich może być uruchomiony na wszystkich komputerach Atari ST, ST^E^, itd., drugi — tylko na maszynach z procesorem MC68030, czyli Atari TT lub Falcon030, zaś ostatni i najszybszy wymaga zaistalowania koprocesora matematycznego (na razie koprocesor montowany jest seryjnie tylko w TT-tkach).

Skoro wiemy już co uruchomić przejdźmy więc do parametrów, z którymi należy wywołać program ray tracera. Oto ich lista:

+? Włączenie danej opcji
–? Wyłączenie opcji

| Nazwa | [Zakres] | Znaczenie |
|---|---|---|
| +i‹filename› | | Nazwa pliku wejściowego zawierającego opis sceny przeznaczonej do obliczeń. Zalecane rozszerzenie .POV |
| +o‹filename› | | Nazwa pliku wyjściowego gdzie ma być zapisany obrazek. Zalecane rozszerzenia w zależności od formatu: targa — *.TGA, dump — *.DIS, raw — *.R8, *.G8, *.B8 |
| +w##### | [1–32767] | Rozdzielczość pozioma obrazu |
| +h##### | [1–32767] | Rozdzielczość pionowa obrazu |
| +v | | Wyświetlanie stanu programu w czasie obliczeń |
| +x | | Przerwanie obliczeń w dowolnym momencie |
| +f(?) | [t, d lub r] | Format pliku wyjściowego, odpowiednio: targa, dump, raw |
| +s##### | [1–32767] | Początkowa linia przy renderowaniu fragmentu obrazu |
| +e##### | [1–32767] | Końcowa linia przy renderowaniu fragmentu obrazu |
| +c | | Kontynuowanie przerwanej poprzednio pracy |
| +p | | Oczekiwanie na naciśnięcie klawisza przy wyjściu do systemu |
| +b##### | [1–1000] | Rozmiar bufora wyjściowego (zalecana wartość 30) |
| +a##### | [0.0–1.0] | Rozmywanie rastra wyjściowego (antyaliasing) |
| +q# | [0–9] | Jakość obrazu: 9 najwyższa (standardowa) — 0 najniższa |
| +l‹path› | | Ścieżka dostępu do katalogu zawierającego pliki biblioteczne (przy czym może być zdefiniowana więcej niż jedna ścieżka dostępu) |

Do czasu wyłączenia generowania pliku wyjściowego opcją –f POV-Ray tworzy obrazek w jednym z następujących formatów:

+ft — nieskompresowana Targa-24 — *.TGA (format standardowy)
+fd — format dump — *.DIS
+fr — format raw — jeden plik na każdą składową koloru R — *.R8 lub *.RED, G — *.G8 lub *.GRN, B — *.B8 lub *.BLU

Opcja +a[###] włącza antyaliasing czyli rozmywanie rastra. Opcja ta jest zalecana przy niskiej rozdzielczości i dużej ilości kolorów. Liczba podana jako paramert wyznacza próg antyaliasingu. Jeżeli kolor punktu różni się od sąsiedniego więcej niż podany próg, to kolor takiego punktu ustawiany jest jako średnia arytmetyczna obu pikseli. Gdy podamy próg 0.0 to każdy punkt obrazka zostanie uśredniony. Polecany zakres wartości — 0.2 do 0.4.

Możemy przyspieszyć nieco proces ray tracingu podając, przy pomocy opcji +b[#####], odpowiedni rozmiar bufora wyjściowego, gdzie umieszczane są obliczone dane obrazu przed zapisem na dysk. Wartość podana jako paramert oznacza wielkość tegoż bufora w kB, jeżeli nie utworzymy bufora to renderowana linia będzie zapisywana zaraz po jej obliczeniu.

Jeżeli realizacja sceny zostanie przerwana przed końcem, to może być ona wznowiona przy pomocy rozkazu +c. Opcja ta odczytuje plik wyjściowy z poprzednimi danymi, wyświetla go na ekranie i rozpoczyna dopisywanie nowych danych.

W przypadku gdy trzeba obliczyć fragment obrazu, można to zrobić podając linie początkową i końcową (odpowiednio: –s### i –e###). Opcja –s### jest ignorowana przy kontynuowaniu raytracingu, automatycznie znajdowana jest ostatnia obliczona linia i od niej kontynuowane są obliczenia.

Opcja –q# specyfikuje jakość renderowanego obrazu, parametr może przybierać następujące wartości:

▷39

Marcin Winiarski

# THE CIVILIZATION

Pomimo bardzo znaczących strat poniesionych przez Chińczyków podczas wojny z Babilończykami na Półwyspie Arabskim, już latem 76 r.p.n.e. byli oni tak silni by wedrzeć się na tereny Cesarstwa Rzymskiego i zadać poważne straty wojskom Juliusza Cezara. Mao Tse Tung chciał za wszelką cenę zdobyć żyzne tereny południowej Azji i bogatą w surowce Wyżynę Irańską, co mogło rozwiązać jego problemy z wyżywieniem stale rosnącej populacji jego państwa, a poza tym doskonale rozwinięte kopalnie, wybudowane niegdyś przez Francuzów na Wyżynie Irańskiej dostarczyłyby niezbędnych surowców i funduszy do rozwoju gospodarki i armii chińskiej. Zadanie jednak nie należało do najłatwiejszych. Cesarstwo Rzymskie nie było co prawda ani zbyt dobrze rozwinięte, ani też nie posiadało silnej i nowoczesnej armii, było jednak w bardzo dobrych układach z Rosjanami, których potęgi militarnej i gospodarczej nie należało ignorować. Już podczas pierwszych starć z flangami rzymskimi ambasador chiński w Moskwie otrzymał ostrzeżenie od władz rosyjskich, że nie będą one tolerowały takiego postępowania i jeśli ataki nie zostaną przerwane, Moskwa przystąpi do wojny. Przerwanie wojny oznaczało dla Chin rezygnację z żyznych ziem i dochodowych kopalni, czyli w niedługim czasie głód oraz katastrofę ekonomiczną, poza tym Rzym na pewno zażąda wysokiej kontrybucji, której Chiny nie będą w stanie zapłacić, co będzie powodem do nowej wojny. Szczęśliwie dla Mao Tse Tunga Rosjanie sąsiadowali z wojowniczymi plemionami Azteków a ciągłe potyczki wojsk na granicy między dwoma państwami nie oznaczały bynajmniej ich wzajemnej, braterskiej miłości. W tej sytuacji chiński dyplomata, który przybył do stolicy państwa Azteków nie musiał wiele się natrudzić, aby przekonać Zulu Czakę o konieczności ostatecznego rozprawienia się z uciążliwym sąsiadem. Dzięki jednemu posunięciu dyplomatycznemu sytuacja zmieniła się diametralnie, Chiny zyskały potężnego sojusznika, który zatrzymał większość sił rosyjskich w południowej części kraju, gdzie stawiały one marny opór dobrze wyszkolonym legionom Azteków. W tej sytuacji Moskwa nie mogła udzielić wystarczającej pomocy militarnej Cesarstwu Rzymskiemu, które słabo przygotowane do obrony, przegrywało kolejne bitwy. W krótkim czasie wojskom Mao Tse Tunga udało się zająć większość miast rzymskich, zrabowano przy tym ogromne ilości złotych monet i wszelkiego rodzaju kosztowności, które zasiliły prawie pusty skarbiec Chin. Padł Konstantynopol i Ateny, główne ośrodki kultury dawnego mocarstwa, w ręce Chińczyków wpadły plany nowoczesnej machiny oblężniczej i to przesądziło o dalszych losach wojny, doskonale ufortyfikowany i przygotowany do obrony Rzym nie oparł się własnej myśli technicznej, padł w ciągu paru dni. Rosjanie, nie chcąc prowadzić wojny na dwa fronty, podpisali z Chinami traktat pokojowy. Mao Tse Tung mógł spokojnie zająć się rozwojem własnego państwa.

Tak potoczyły się losy Cesarstwa Rzymskiego i Juliusza Cezara pewnego pięknego sierpniowego wieczoru 1993 roku. Podczas innych wieczornych sesji bywało, że Napoleon pod Waterloo ulegał władcy Mongołów, Egipcjanie rozpoczynali pierwszą wojnę nuklearną, Niemcy stawiali piramidy, a Babilończycy dokonywali podboju kosmosu. A wszystko to stało się możliwe dzięki doskonałemu produktowi firmy Microprose, grze Civilization. Gra ta stanowi przełom w produkcji gier strategicznych. Dotychczas głównym zadaniem gracza było zniszczenie przeciwnika, a głównym środkiem do osiągnięcia tego celu była własna armia.

Civilization wymaga od Ciebie zupełnie czegoś innego, nie będziesz już musiał z uporem maniaka niszczyć wszystkich jednostek, które różnią się od Twoich kolorem, tu Twoim głównym celem jest zapewnienie rozwoju własnej cywilizacji. Musisz więc nie tylko okazać się dobrym strategiem, ale również, a może przede wszystkim, sprytnym dyplomatą, mądrym ekonomistą, przewidującym nastroje społeczne władcą, bystrym politykiem i błyskotliwym architektem (musisz zaprojektować wygląd Twojego pałacu). No ale zacznijmy od początku. Po włączeniu gry na ekranie ukażą się następujące komunikaty:

**Start a New Game:**

Po wybraniu tej opcji będziesz musiał wybrać poziom, na którym zechcesz grać — od *chieftain* (najprostszy), do *emperor* (najtrudniejszy), początkującym graczom radzę wybrać ten pierwszy, aby nie okazało się, że w momencie kiedy wynajdziecie koło, ktoś zaatakuje Was głowicami nuklearnymi. Później należy wybrać ilość cywilizacji, które wezmą udział w grze, od 3 do 7. Najlepiej wybrać 7 — więcej możliwości pertraktacji, kontaktów, intryg, a przede wszystkim łatwiej podbić mniejszego wroga. Następną czynnością jest wybranie z pośród 14 cywilizacji tej, którą będziesz dowodził. Amerykanie — wbrew pozorom — nic specjalnego, Grecy — dość dobrze żeglują, nic poza tym, Rzymianie — dość zadziorni, jeśli zostaną Twoim sąsiadem, uważaj na nich, Babilończycy — doskonali naukowcy, jeśli ich wybierzesz, wynalazków Ci nie zabraknie, a nauka będzie stała na bardzo wysokim poziomie, Niemcy, Rosjanie, Zulusi, Aztekowie — o tych cywilizacjach można powiedzieć jedno: miej się na baczności jeżeli zostaniesz ich sąsiadem (na pewno Cię zaatakują), wybierając jedną z nich możesz wyeliminować groźnego przeciwnika (sam nim będziesz), Egipcjanie, Indianie, Anglicy — nic specjalnego, Francuzi, Chińczycy, Mongołowie — dość zadziorni, ale mniej niebezpieczni od Niemców, Rosjan i Azteków.

**Load a Saved Game:**

Po wybraniu tej opcji możesz odtworzyć jedną z wcześniej zapisanych gier.

**EARTH:**

Pokazuje i opisuje historię powstania Ziemi.

**Customize World:**

Ta opcja pozwoli Ci na określenie wielkości kontynentów i warunków naturalnych na nich panujących.

**View Hall of Fame:**

Opcja pozwala na przeglądnięcie listy punktów uzyskanych przez poszczególne cywilizacje.

Po uporaniu się ze wszystkimi formalnościami rozpoczynasz grę. Na początku masz do dyspozycji *settlers* czyli osadników, mogą oni między innymi budować miasta — *(Found New City)*. Twoim pierwszym zadaniem będzie znalezienie odpowiedniego miejsca na miasto, musisz przede wszystkim pomyśleć o tym, aby na około niego były dobre gleby, to zaoszczędzi Ci w przyszłości problemów z wyżywieniem. Najlepiej jeśli miasto jest przy rzece lub gdy ta płynie przez miasto, dobrym otoczeniem są też łąki. Drugim ważnym elementem jest obronność miasta: jeśli w jego otoczeniu znajdują się góry lub lasy, będzie to dość znaczącym utrudnieniem dla przeciwnika, zamierzającego je podbić. Kiedy już znajdziesz odpowiednie miejsce, wybierz *„Found New City"* z menu *„ORDERS"*. Twoje pierwsze miasto będzie jednocześnie stolicą Twego państwa, jeśli jednak będziesz chciał przenieść stolicę do innego miasta, wystarczy wybudować w nim pałac. Każde z Twoich miast będzie wystawiać jednostki wojskowe *(military units)*, oddziały osadników *(settlers)* oraz rozwijać się budując różnego rodzaju budowle *(city improvements)*. Nowo wybudowane miasto należy przede wszystkim zabezpieczyć przed atakami, musisz więc wystawić jednostkę wojskową, która będzie je broniła.

W początkowej fazie gry do dyspozycji będziesz miał jedynie *„Militia"*, oddziały łuczników, nadające się świetnie do obrony miast. W miarę jak będzie rozwijała się nauka do swojej dyspozycji będziesz miał coraz lepsze oddziały. Następnie możesz wybudować jakiś budynek użyteczności publicznej np. *granary* — spichlerz, *marketplace* — centrum handlu czy *city walls* — mury obronne. W zależności od rodzaju budynku jaki wybudujesz miasto będzie przynosić większy dochód, szybciej produkować wszelkiego rodzaju urządzenia lub też będzie bardziej odporne na ataki. Bardzo ważne jest, aby zarówno miasta były dobrze chronione, jak i aby mogły wystarczająco szybko produkować różnego rodzaju dobra, dlatego też nie można skupić się jedynie na produkcji machin wojennych i wystawianiu coraz to nowych oddzia-



łów wojsk. Taka polityka to działanie na krótką metę. Następnym ważnym czynnikiem prawidłowego rozwoju Twoich miast jest wybudowanie odpowiedniej infrastruktury wokół nich. Chodzi tu przede wszystkim o zwiększenie wydajności pól uprawnych, znajdujących się w pobliżu miasta oraz wykorzystanie surowców naturalnych (węgiel, złoto). Do tego celu służą oddziały *settlers* (osadników), które mogą budować *irrigation* (systemy nawadniające) oraz *mines* (kopalnie). Systemy nawadniające należy budować w pobliżu źródeł wody i łączyć ze sobą, w celu doprowadzenia wody do oddalonych pól. Kopalnie przynoszą zyski jeśli są wybudowane w górach lub na terenie pagórkowatym, ale największe dochody przynoszą kiedy na ich obszarze znajduje się złoto lub węgiel. Postaraj się więc o zdobycia terenów zasobnych w te bogactwa. Aby Twoje kopalnie i pola prawidłowo funkcjonowały, powinieneś połączyć je z miastem drogami, do ich budowy używaj oddziałów *settlers*, wybierając z menu *ORDERS* podpunkt *roads* (drogi). Drogami powinieneś też połączyć własne miasta, to ułatwi się między nimi oddziałów wojskowych, jest to bardzo przydatne, kiedy musisz szybko skoncentrować w jednym mieście większe siły wojska, w celu jego obrony lub stłumienia buntu mieszkańców. Drogi ułatwiają też handel między państwami. Do obrony ważnych strategicznie miejsc, granic państwa, szlaków handlowych możesz wykorzystać *fortress* (fortecę). Są one budowane przez oddziały *settlers* i zwiększają możliwości obronne jednostek, które znajdują się w ich obrębie o 100 procent. Miasta są też źródłem siły militarnej, mogą one wystawiać oddziały wojskowe. Ilość dostępnych w grze rodzajów sił zbrojnych zwiększa się wraz z rozwojem nauki i techniki. Wszystkie rodzaje wojsk można podzielić na trzy grupy: oddziały służące do obrony (*phalanx* — flanga, *musketeers* — muszkieterowie, *mech. inf.* — piechota zmechanizowana, itp.), Oddziały o dużej sile ataku (*cavarly* — kawaleria, *cannon* — artyleria, *bomber* — bombowce, itp.), oddziały, które zapewnią Ci potęgę na morzu (*ironclad* — parowiec, *carrier* — lotniskowiec, itp.). Poza tym są jeszcze specjalne jednostki, które służą do odkrywania nowych kontynentów, transportu wojsk, kolonizacji nowych planet. Wojnę można rozpocząć bezpośrednio atakując jednostkę przeciwnika; do ataku należy używać wojsk o dużej sile uderzeniowej. Jeśli jednostki przeciwnika są ufortyfikowane (wokół znaku jednostki jest ciemniejsza ramka) ich siła obronna wzrasta o 50 procent. Jeżeli przeciwnik znajduje się w fortecy, nie należy liczyć na to, że uda się go zniszczyć jedną jednostką, no chyba, że nasza technika wojenna stoi na znacznie wyższym poziomie i atakujemy legion bombowcem. W innej sytuacji najlepiej jest zgromadzić trzy jednostki, które będą atakować fortecę. Nie należy jednak zapominać, że liczba znajdujących się w twierdzy oddziałów przeciwnika oraz ich typ nie jest nam znany — widzimy tylko jedną jednostkę, a w rzeczywistości może ich być kilka. Miasta najlepiej atakować mając do dyspozycji artylerię lub katapulty, w późniejszej fazie gry do tego celu można też używać bombowców, panceników i ciężkich czołgów. Dobrym posunięciem taktycznym jest obleganie miast. Można w tym celu używać wszystkich jednostek wojskowych, należy przy tym ufortyfikować własne jednostki, wybierając z menu *ORDERS* podpunkt *fortyfy*, co spowoduje wzrost ich siły obronnej o 50 %. W przypadku gdy przeciwnik zaryzykuje wypad za mury miasta — Ty będziesz na to dobrze przygotowany. Poza tym szczelne obleganie uniemożliwi przeciwnikowi dostarczenie pomocy militarnej. Aby zmniejszyć potencjał produkcyjny miasta, jego siłę i wielkość, należy zniszczyć jego infrastrukturę, można to zrobić przy pomocy wszystkich jednostek, wybierając z menu *ORDERS* podpunkt *pillage* (grabież, rabunek). Jeśli atak się powiedzie w Twoje ręce wpadnie nie tylko miasto, ale całe złoto, które się w nim znajduje i wszystkie zdobycze techniczne danej cywilizacji. Moment zdobycia miasta jest też momentem przełomowym w wojnie. Twój przeciwnik, w tym



momencie, bardzo często proponuje pokój. Ty ze swojej strony możesz zażądać zapłacenia wysokiej kontrybucji. Istnieje też zupełnie inny sposób prowadzenia wojen i zdobywania miast: jest nim dyplomacja. Dyplomata może przekupywać jednostki przeciwnika, co powoduje, że przechodzą one na Twoją stronę. Jest to bardzo użyteczne, jeśli zależy Ci na zdobyciu miejsc dobrze ufortyfikowanych i ważnych strategicznie punktów. Poza tym dyplomata może założyć ambasadę. W tym celu należy wejść do miasta jakiejś innej cywilizacji i wybrać opcję *Establishment Embassy*. Po założeniu ambasady, w menu *ADVISORS* i podpunkcie *Intelligence Advisor*, pojawią się dane o państwie, w którym założono ambasadę: ilość wojsk, jakimi dysponuje, ustrój polityczny, ilość pieniędzy w skarbcu, wykaz osiągnięć techniki, itp. Posiadanie ambasady umożliwi też rozbudowę siatki szpiegowskiej w danym kraju. Od tej pory będziesz wiedział o wszystkich posunięciach przeciwnika. Dyplomata może też ukraść nowoczesną technologię (*steal technology*) co umożliwi Ci jej wykorzystanie do własnych celów. Uwaga: Kradzież nie popłaca; przeciwnik bardzo często w takiej sytuacji wypowiada wojnę. Można też, przy pomocy dyplomaty, sprawdzić jak wielkie jest miasto przeciwnika, ile, i jakiego rodzaju, jednostki się w nim znajdują, co zostało w nim zbudowane, ilu ludzi w nim zamieszkuje, jakie są ich nastroje, itd. Służy do tego opcja *Investigate City*. Aby zdobyć miasto bez walki i zniszczeń nią wywołanych, należy użyć opcji *Incite a Revolt*. Twój dyplomata przekupi ludność znajdującą się w mieście, jeśli oczywiście dysponujesz odpowiednią sumą pieniędzy. Ten sposób prowadzenia walki jest wyjątkowo szybki i efektywny. W Twoje ręce wpada niezniszczone miasto — nie musisz tracić czasu i pieniędzy na jego odbudowę. Dyplomata przydaje się również kiedy chcesz przeprowadzić sabotaż w celu zniszczenia nowo powstających budowli w mieście (*Industrial Sabotage*). Jeśli chcesz przeprowadzić rozmowy z obcym władcą w celu podpisania traktatu pokojowego, uzyskania kontrybucji, pomocy militarnej, nowych technologii, itp. wybierz opcję*Meet with King*.

Bardzo ważnym czynnikiem prawidłowego rozwoju państwa jest rozwój nauki i odpowiedni system rządów. Dokładnie o tych sprawach następnym razem, a poza tym: dokładne dane o rodzajach wojsk, ich sile i przeznaczeniu, dokładny opis budowli miejskich (do czego służą, kiedy należy je stawiać), opis wszystkich wynalazków i ich cel.

**Dane techniczne**:

Wyszczególnione tu dane techniczne dotyczą rodzajów podłoża, spotykanych w grze oraz ilości żywności i pieniędzy, jaką można otrzymać z danego typu obszaru.

„Produkcja żywności" oznacza ilość żywności jaką można uzyskać z danego gruntu (od 1 do 4). Może ona się zmieniać w zależności od tego, czy na danym obszarze jest system nawadniający czy go nie ma oraz od rodzaju zwierzyny tam występującej. Ilość pieniędzy, którą można uzyskać z danego obszaru zależy od tego czy ma on jakieś dobra, które można eksploatować i czy można prowadzić na jego obszarze jakąś produkcję („współczynnik produkcji"). Im większy „współczynnik produkcji" tym większy dochód. „Współczynnik produkcji" można zwiększać stawiając na danym obszarze kopalnie. Tereny górzyste lub pagórkowate mogą utrudniać manewry wojsk, dlatego jednostki, twierdze, miasta, które znajdują się na ich terenie są mniej narażone na ataki ze strony wrogów. Jak wzrasta obronność tych jednostek na danym obszarze jest podane procentowo.

**Arctic** — lądolód, produkcja żywności = 0 (jeśli występuje z wizerunkiem foki = 2), o 0 % wzrasta Twoja obronność.

**Desert** — pustynia, współczynnik produkcji = 1 (z kopalnią = 2), produkcja żywności = 0 (z systemem nawadniającym = 1, jeśli na pustyni znajduje się oaza = 4 oraz system nawadniający = 5), o 0 % wzrasta obronność.

**Forest** — las, produkcja żywności = 1, współczynnik produkcji = 2, o 50 % wzrasta obronność.

**Grassland** — łąka, produkcja żywności = 2 (z systemem nawadniającym = 3), współczynnik produkcji = 0, o 0 % wzrasta obronność.

**Hills** — wzgórza, produkcja żywności = 1 (z systemem nawadniającym = 2), współczynnik produkcji = 0 (z kopalnią = 3, jeśli na danym obszarze występuje węgiel = 5), o 100 % wzrasta obronność.

**Jungle** — dżungla, produkcja żywności = 1, współczynnik produkcji = 0, o 50 % wzrasta obronność.

**Mountains** — góry, współczynnik produkcji = 1 (z kopalnią = 2), produkcja żywności = 0, o 200 % wzrasta obronność.

**Ocean** — ocean, produkcja żywności = 1, współczynnik produkcji = 0, o 0 % wzrasta obronność.

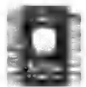
**Plains** — równiny, produkcja żywności = 1 (z systemem nawadniającym = 2), współczynnik produkcji = 1 (jeśli na danym obszarze występują konie = 3), o 0 % wzrasta obronność.

**River** — rzeka, produkcja żywności = 2 (z systemem nawadniającym = 3), współczynnik produkcji = 0, o 50 % wzrasta obronność.

**Swamp** — bagna, produkcja żywności = 1, współczynnik produkcji = 4, o 50 % wzrasta obronność.

**Tundra** — tundra, produkcja żywności = 1, współczynnik produkcji = 0, o 0 % wzrasta obronność.

❑



Witold Warczak

# Szara skrzynka cz. 3

Witam wszystkich wytrwałych w trzecim odcinku cyklu „Szara skrzynka". Zgodnie z zapowiedzią w artykule tym dokończymy opis systemu przerwań, a także zajmiemy się układem zarządzającym pamięcią i systemem wyświetlania obrazu. Wszystkich zniecierpliwionych sporą dawką teoretycznych informacji proszę jeszcze o odrobinę wytrwałości. Mogę obiecać, że już w następnym numerze pojawią się gotowe procedury, pozwalające wykorzystać „ukryte" możliwości komputerów ST/ST$^E$. Niestety, aby móc zrozumieć ich działanie i w pełni je wykorzystać, trzeba najpierw dobrze poznać sprzęt na którym się pracuje. A więc zaczynamy...

**System przerwań — ciąg dalszy**

Pozostaje jeszcze problem jak wykorzystać przerwania do własnych celów. Aby tego dokonać należy:

1. Napisać własną procedurę obsługi przerwania,

2. Odpowiednio zmienić adres wektora (oraz ewentualnie zaprogramować MFP),

3. Ustawić poziom przerwań w masce przerwań procesora. Od tego momentu zacznie się wykonywać nasza procedura przerwania.

Adresy wektorów i poziomy przerwań oraz rejestry układu MFP zostały omówione w ST$^E$fanie 3,4/93, więc pozostał jeszcze problem maski przerwań. Aby zezwolić na interesujące nas przerwania należy wpisać do rejestru SR procesora (dostęp jedynie w trybie supervisor) jedną z następujących wartości:

| Wartość | Przerwania obsługiwane |
|---|---|
| *$2100* | *wszystkie (MFP+VBL+HBL)* |
| *$2300* | *MFP+VBL* |
| *$2500* | *MFP* |
| *$2700* | *przerwania zabronione* |

Oczywiście pod pojęciem MFP należy rozumieć wszystkie przerwania jakie jest w stanie generować ten układ, tworzy on bowiem własne wektory przerwań.

Zobaczmy jak instalować własne przerwania:

**Przerwanie HBL**

Jak wiesz przerwanie to jest wywoływane gdy zakończy się rysowanie każdej z 313 (telewizor) poziomych linii ekranu (wraz z ramką). W celu jego uaktywnienia wystarczy włączyć tryb supervisor (wszystkie rejestry sprzętowe są dostępne tylko w tym trybie), zablokować przerwania (aby uniknąć przypadkowego uruchomienia procedury po zmianie wektora), wpisać adres nowej procedury (długie słowo) do komórki *$68* i ustawić poziom przerwań na *$2100*. Tak to wygląda teoretycznie. W praktyce nie jest tak prosto (przynajmniej na moim sprzęcie — 1040 ST$^E$ i TOS 1.061), gdyż systemowa procedura przerwania VBL blokuje przerwania HBL (nie używane przez system) ustawiając maskę przerwań na *$2300*. Istnieje proste wyjście z tego kłopotu — wystarczy pozbyć się systemowej procedury przerwania VBL (problemy zaczynają się gdy potrzebujemy systemowego VBL'a). Całość przedstawia listing HBL.S zamieszczony na dyskietce ST$^E$fana. Po uruchomieniu programu na ekranie zobaczysz tęczowe płynące paski — efekt modyfikacji co każdą linię ekranu rejestru koloru tła. Jednak zapewne zauważysz również, że przejścia kolorów czasami drżą, reagując chociażby na ruch myszy, czy naciśnięcie klawisza. Dzieje się tak dlatego, gdyż przerwanie HBL ma najniższy priorytet i może być zakłócone przez inne przerwania. Pisząc własne procedury trzeba o tym pamiętać, ważne jest też aby procedura HBL była jak najkrótsza — wykonuje się ona 15650 razy na sekundę!

**Przerwanie VBL**

Istnieją trzy sposoby instalacji własnej procedury tego przerwania:

Pierwszy z nich podobny jest do tego jak zostało to zrobione przy omawianiu przerwania HBL. Adres naszej procedury przerwań (długie słowo) wpisujemy do komórki o adresie *$70* (oczywiście warto wcześniej zablokować przerwania, a także zapamiętać adres systemowej procedury VBL, żeby móc ją przywrócić kończąc program). Od momentu zezwolenia na przerwania VBL (maska równa *$2100* lub *$2300*) zacznie się wykonywać nasza procedura. Takie rozwiązanie nie zawsze jest najwygodniejsze — tracimy wówczas np. systemową procedurę obsługi myszy.

Drugi sposób pozwala na częściowe uniknięcie tego kłopotu. Polega on na podstawieniu własnej procedury przerwania (podobnie jak to zostało zrobione poprzednio), lecz zakończonej zamiast rozkazu RTE skokiem do procedury w ROM-ie. Uwaga: adres skoku należy odczytać z komórki o adresie *$70* przed podstawieniem własnej procedury, a nie posługiwać się adresem znalezionym dla konkretnej wersji ROM-u. Niestety ten sposób może być kłopotliwy dla początkujących programistów a poza tym nieelegancki (wymaga użycia kodu samomodyfikującego się — brrr...) i jest sens go stosować tylko wtedy gdy nasza procedura musi się wykonywać jako pierwsza (przed systemową).

Trzeci sposób instalacji własnego VBL'a jest najbardziej elegancki i legalny. Twórcy systemu operacyjnego przewidzieli możliwość instalacji własnych procedur przerwań i zastosowali coś co się nazywa kolejką przerwań. Zmienna systemowa *vblqueue ($456)* zawiera adres tablicy wektorów do procedur wykonywanych w czasie przerwania VBL. Liczba tych procedur wynosi 8, ale w razie potrzeby może być zmieniona. W wypadku gdy chcemy zainstalować swoją procedurę do kolejki przerwań robimy to następująco: odczytujemy adres tablicy wektorów z *vblqueue*, następnie sprawdzamy kolejno długie słowa (4 bajty) począwszy od odczytanego adresu. Gdy napotkamy wartość 0 wpisujemy tam adres naszej procedury, która zacznie się wykonywać najpóźniej po 1/50 sekundy od momentu wpisania adresu. W wypadku gdy przekroczymy długość tablicy (8 długich słów) możemy skopiować ją w inne miejsce pamięci zmienić adres w *vblqueue* i liczbę procedur w *nvbls ($454)*. Chcąc usunąć jakąś procedurę z kolejki przerwań zamiast jej adresu należy wpisać wartość 0. Tak zainstalowana procedura powinna być zakończona rozkazem RTS zamiast RTE.

Wybór sposobu należy do Ciebie, w demosach zwykle stosuje się sposób pierwszy, w grach pierwszy i drugi, a w programach użytkowych trzeci. Przykłady instalacji przerwań VBL pokazane są na listingach VBL1.S i VBL2.S na dyskietce ST$^E$fana.

Na koniec kilka uwag praktycznych. Przerwanie VBL często wykorzystywane jest do różnorodnych operacji na pamięci ekranu, takich jak scrolle, sprite'y, itp. Korzysta się przy tym z faktu, że w momencie wywołania przerwania nie jest tworzony obraz. Wszystko się zgadza, ale trzeba pamiętać, że w przypadku dłuższej procedury obsługi przerwanie VBL może się nie zakończyć przed ponownym rozpoczęciem wyświetlania. Wtedy wszelkie modyfikacje pamięci ekranu mogą powodować widoczne zakłócenia. W niektórych wypadkach (wszelkiego rodzaju tęcze, syncscrolle, overscany, itp.) konieczna jest dokładna synchronizacja wykonywania przerwania VBL z wyświetlaniem obrazu. Niestety, kolejne przerwania nie występują w idealnie dokładnych odstępach czasu (spowodowane jest to odświeżaniem pamięci, działaniem DMA, itp.). Aby uzyskać dokładną synchronizację wystarczy umieścić w procedurze przerwania VBL króciutką procedurkę:

```
vbl: MOVE.W #$2700,SR ;blokujemy
     wszystko
     ;tutaj inne procedury (o stałym
     czasie wykonywania)
loop: MOVE.B $FFFF8209,d0
     BEQ loop
     NOT.b D0
     LSL.B D0,D0
     ;tutaj procedura, która zacznie
     się wykonywać synchronicznie z
     wyświetlaniem obrazu
     RTE
```

Działa toto następująco: Najpierw sprawdza jaka jest rozbieżność pomiędzy wyświetlaniem obrazu, a realizacją procedury przerwań, korzystając z rejestru licznika vcontl (co to jest patrz dalej). Następnie wykorzystywany jest fakt, że czas wykonywania rozkazu *LSL.B D0,D0* jest zależny od zawartości rejestru D0. Zapewnia to synchronizację wykonywania dalszych rozkazów z zawartością rejestru *vcontl*, a więc z pozycją strumienia elektronów na ekranie. Dzięki temu, rozkazy następujące po tej procedurce będą się wykonywać zawsze w tym samym momencie. Procedurka ta została wymyślona (prawdopodobnie) przez grupę TEX i jest spotykana w różnych programach.

**Przerwania generowane przez MFP**

Ta grupa przerwań jest najtrudniejsza do wykorzystania. Przyjrzyjmy się jak wygląda programowanie liczników MFP. Przypominam, że rejestry MFP zostały omówione w poprzednim odcinku „Szarej skrzynki" — ST$^E$fan 3,4/93.

Na początek zajmiemy się korzystaniem z zegarów MFP w trybie zliczania impulsów. Jak pamiętamy na wejście zegara B podłączany jest sygnał DE. Sygnał ten zmienia się ze stanu wysokiego na niski po zakończeniu kreślenia każdej z 200 (monitor kolor) lub 400 (monitor mono) widocznych linii ekranu w momencie gdy zaczyna się wyświetlanie bocznej ramki. Pozwala to na wywołanie przerwania po zakończeniu wyświetlania określonej linii ekranu. Spróbujmy zatem napisać program, który w linii 150 zmienia kolor ekranu na czarny. W tym celu należy najpierw zaprogramować układ MFP. Na początek wyłączamy zegar B, aby uniknąć przypadkowego wywołania przerwania. W tym celu wpisujemy do rejestru *TBCR* wartość 0. Ponieważ w przypadku sygnału DE istotne znaczenie ma poziom, ustawiamy w rejestrze *AER* reakcję na zbocze opadające. Ponadto pracować będziemy w trybie programowego zakończenia obsługi przerwania (nie jest to konieczne, ale warto wiedzieć jak sobie z tym radzić). Następnie wpisujemy do rejestru danych *(TBDR)* numer linii, w której ma zostać wywołane przerwanie, oraz ustawiamy w rejestrach *IMRA* i *IERA* bit zezwalający na przerwania zegara B. Teraz jeszcze jedna sprawa. Zegar ten musi startować synchronicznie z wyświetlaniem obrazu, dlatego też programując MFP wpisaliśmy do rejestru *TBCR* wartość zero, czyli stop. Aby uzyskać synchronizację zegara z obrazem będziemy go uruchamiać na przerwaniu VBL, wpisując do rejestru *TBDR* numer linii, w której zegar ma zgłosić przerwanie i do rejestru *TBCR* wartość 8, czyli start zegara w trybie zliczania. Instalacja procedury przerwania VBL przebiega w sposób opisany powyżej. Procedura obsługi przerwania zegara B ładuje do rejestru sprzętowego określającego kolor tła wartość 0 (czarny), zatrzymuje licznik B i informuje MFP o zakończeniu obsługi przerwania korzystając z rejestru *ISRA*. I to by było na tyle. Jak sprawa wygląda w postaci programu możesz się dowiedzieć analizując listing MFP1.S na dyskietce ST$^E$fana.

Nieco inaczej wygląda programowanie MFP w trybie opóźniania. Tryb ten jest często używany na przykład do odtwarzania sampla, gdzie potrzebne jest przerwanie o określonej częstotliwości. Zaletą tego trybu jest możliwość generowania przerwań z dowolną częstotliwością od kilku Hz do około 30 Khz. Przypuśćmy, że chcemy uzyskać przerwanie zegara A z częstotliwością 17 Khz. Musimy zatem odpowiednio zaprogramować rejestry *TADR* i *TACR* układu MFP. W rejestrze *TACR* wybieramy tryb pracy zegara (opóźnianie) i współczynnik podziału. Współczynnik podziału należy dobrać tak, aby uzyskać maksymalną dokładność ustawienia częstotliwości. Dla dużych częstotliwości (od 10 kHz) zwykle wybiera się podział przez 4. Do rejestru danych zegara A wpisujemy wartość wyliczaną z następującego wzoru:

*data=Fd/Fp*

gdzie

*Fp* — żądana częstotliwość przerwań

*Fd* — częstotliwość zliczania wynikająca ze wzoru:

*Fd=2457/podział*

gdzie *„podział"* jest współczynnikiem podziału wynikającym z zawartości rejestru *TACR*. Tak więc chcąc uzyskać częstotliwość przerwań 17 kHz otrzymujemy:

*Fd=2457/4=614*

*data=614/17=36.1*

Tak więc do *TACR* wpisujemy 1 (opóźnianie, podział przez 4), zaś do *TADR* wpisujemy 36. Po ustawieniu zawartości tych rejestrów pozostaje jeszcze wyzerować bit 3 w rejestrze *VR* (automatyczne zakończenie obsługi przerwania) oraz zezwolić na przerwania w rejestrach *IERA*, *IMRA*. Niestety, tym razem przykładowy listing (plik MFP2.S) ilustruje tylko zasadę, gdyż odtwarzanie sampli to temat na bliską przyszłość.

Pozostał jeszcze tryb pomiaru, lecz ma on stosunkowo małe zastosowanie praktyczne, więc nie będzie on szczegółowo omawiany.

Odnośnie przerwań to już wszystko. Zdaję sobie sprawę że jest to nieco zagmatwane i trudne do zrozumienia, lecz naprawdę warto się bardziej zainteresować przerwaniami. W razie większych niejasności proszę o listy.

**Zarządzanie pamięcią**

Za zarządzanie pamięcią Atari odpowiedzialny jest układ MMU w serii ST lub układ GST MCU w komputerach serii ST$^E$. Dokładniej działanie tego układu zostało opisane w ST$^E$fanie 1,2/93, dlatego tutaj zajmiemy się tylko kilkoma drobiazgami. Ponieważ mikroprocesor MC 68000 ma 24-bitową szynę adresową, a wewnętrzne rejestry adresowe 32-bitowe, najstarszy bajt 32-bitowego adresu jest ignorowany, a więc wynika z tego, że może on zaadresować 16 MB pamięci. Obszar ten został podzielony w sposób następujący:

*$00000000–$00000800* — obszar wektorów i zmiennych systemowych, dostępny w trybie supervisor.

*$00000800–$003FFFFF* — obszar 4 MB RAM dla użytkownika.

*$00FA0000–$00FFFFFF* — pamięć ROM, rejestry układów sprzętowych, dostęp w trybie supervisor.

Dokładniejsza mapa pamięci przedstawiona została w pierwszym odcinku „Szarej skrzynki" (ST$^E$fan 1,2/93). Jak widać pomiędzy adresami *$3FFFFF–$FA0000* jest spora dziura — zapełniają ją niektóre rozszerzenia (np. do 12 MB). Mimo, iż do adresowania pamięci wystarczą trzy bajty, to jednak dla zgodności z MC 68020 i 68030 powinno się używać pełnych 32-bitowych adresów. Przy okazji wypada wyjaśnić jeszcze pojęcie najmłodszego, średniego, starszego i najstarszego bajtu. Jeżeli mamy 32-bitową liczbę (np. adres) to:

bity 0–7 — najmłodszy bajt } młodsze słowo
8–15 — średni bajt

15–23 — starszy bajt } starsze słowo
24–32 — najstarszy bajt

Po tych wyjaśnieniach wracamy do problemu zarządzania pamięcią. Cztery megabajty RAM które potrafi obsłużyć MMU zostały podzielone na dwa banki. Każdy z banków może zawierać 128, 512 lub 2048 kilobajtów pamięci, zaś fizycznie w Atari ST jest to 16 „kości" (gdyż szyna danych ma szerokość 16 bitów), odpowiednio po 64 kilobity, 256 kilobitów i 1 megabit (64k x1, 256k x1, 1m x1) albo 2 lub 4 moduły SIMM w Atari ST$^E$ (odpowiednio: 256k x9, 256k x8, 1m x9, 1m x8). W zależności od obsadzenia banków można dostępne są konfiguracje: 256 KB (prehistoria), 512 KB (na wymarciu), 1 MB, 2 MB, 2,5 MB i 4 MB. Pojemność pamięci jest rozpoznawana programowo w czasie startu komputera. MMU posiada rejestr memconf, w którym przechowywana jest informacja o aktualnej konfiguracji pamięci.

*memconf $FFFF8001 xxxxBBAA*

gdzie bity: AA (bank 0)

00 — 128 KB

01 — 512 KB

10 — 2 MB

11 — kombinacja zarezerwowana

bity BB mają identyczne znaczenie i odnoszą się do banku 1

Niestety istnieją trzy wersje MMU. Dwie z nich (montowane w Mega ST i niektórych 1040 ST) mają kłopoty z obsługą niektórych konfiguracji pamięci. Ponadto GST MCU w Atari ST$^E$ nie obsługuje poprawnie konfiguracji 2,5 MB, ale problem ten można obejść programowo.

**Grafika**

Na początek trochę o telewizji. Dlaczego o tym piszę — na identycznej zasadzie działają monitory komputerowe (również Atari SM 124), zaś cała praca komputera jest podporządkowana procesowi wyświetlania. Przyjrzyjmy się jak działa nasz ST/ST$^E$ podłączony do telewizora i pracujący w niskiej rozdzielczości. Całkowite wymiary ekranu (razem z ramkami) to około 420 na 313* linii. Jak zapewne wiesz, obraz na w telewizorze jest tworzony przez strumień elektronów. Wiązka elektronów startuje od górnego lewego rogu ekranu i kreśląc punkty porusza się poziomo w prawą stronę. Na początku każdej poziomej linii rysowana jest lewa ramka (około 50 punktów). W tym czasie sygnał DE (Display Enable) ma poziom niski (nieaktywny). Następnie zmienia on stan na wysoki i rysowany jest widoczny obraz (320 punktów). Po narysowaniu widocznej linii sygnał DE ponownie staje się nieaktywny, co oznacza rysowanie prawej ramki. Gdy strumień elektronów osiągnie prawy brzeg ekranu jest on wygaszany i szybko przeskakuje z powrotem na lewy brzeg ekranu, jednak o linię niżej. Czas, w którym wiązka powraca z prawego na lewy brzeg ekranu, nazywamy powrotem poziomym, zaś częstotliwość z jaką się to dzieje — częstotliwością linii. W czasie wygaszania poziomego generowane jest przerwanie HBL. Proces ten powtarza się przy kreśleniu kolejnych linii. Po narysowaniu całego ekranu wiązka elektronów znajduje się w dolnym prawym rogu ekranu, gdzie jest wygaszana i przeskakuje w górny lewy róg. Czas ten nazywamy powrotem pionowym, zaś częstotliwość powrotów pionowych to częstotliwość ramki. W czasie powrotu pionowego generowane jest przerwanie VBL. Cały proces tworzenia ekranu powtarza sie. Aha, jeszcze jedno: podczas wyświetlania pierwszych (około 56) i ostatnich (około 57) poziomych linii ekranu tworzona jest górna i dolna ramka, a sygnał DE jest nieaktywny. Mam nadzieję, że teraz wszyscy już wiedzą co to jest przerwanie VBL, HBL, oraz dlaczego po zaprogramowaniu MFP tak, aby reagowało na zmianę poziomu sygnału DE możemy mieć dodatkowe przerwanie na początku (reakcja MFP na zbocze narastające) lub na końcu (reakcja MFP na zbocze opadające) każdej widocznej (tzn. nie będącej górną lub dolną ramką) linii ekranu. Schematycznie zostało to pokazane na rysunku 1. Warto zauważyć jeszcze jedną rzecz — jeśli zmienimy coś w pamięci ekranu w czasie przerwania VBL nie będzie to powodowało zakłóceń, gdyż obraz nie jest w tym momencie wyświetlany. Odwrotnie, jeżeli zsynchronizujemy procesor tak, aby modyfikował zawartość tej komórki pamięci ekranu, która jest aktualnie wyświetlana, możemy uzyskać np. efekt przedstawiony w rubryce FX w tym numerze. Wracajmy jednak do spraw grafiki.

Możliwości graficzne Atari ST/ST$^E$ nie są dzisiaj może imponujące, ale dobry programista przy sporej znajomości sprzętu może osiągnąć naprawdę niezłe efekty. Za grafikę naszego komputera odpowiedzialne są trzy główne „kości":

MMU — adresuje pamięć ekranu i wysyła dane do SHIFTER,

GLUE — generuje sygnały synchronizacji,

SHIFTER — zamienia dane odczytane z pamięci na sygnały dla przetworników RGB.

W Atari ST$^E$ dwie pierwsze „kości" zawierają się w GST MCU a ostatnia nosi nazwę GST SHIFTER. Działanie tych „kości" zostało opisane w ST$^E$fanie 1,2/93.

Atari ST/STE mogą wyświetlać obraz (bez specjalnych sztuczek) w 3 rozdzielczościach:

— niskiej 320 x 200 w 16 kolorach na monitorze kolorowym lub telewizorze,

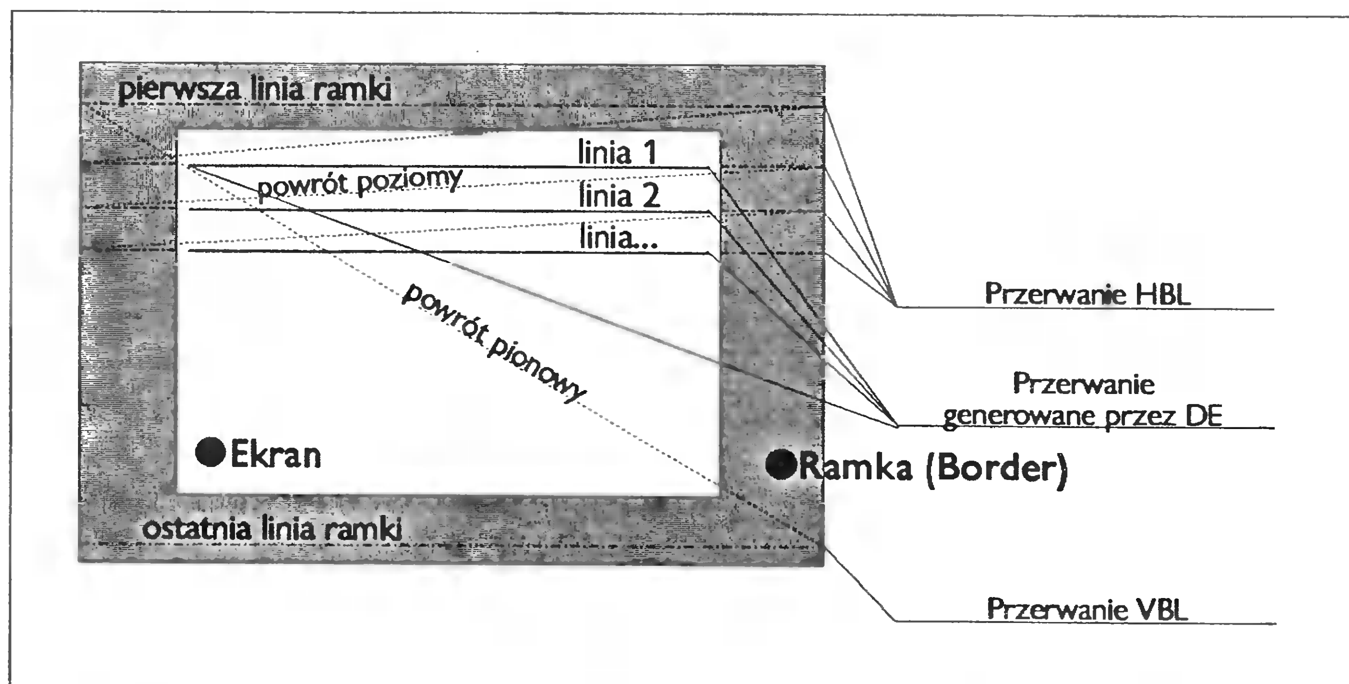
— średniej 640 x 200 w 4 kolorach na monitorze kolorowym lub telewizorze,

— wysokiej 640 x 400 na specjalnym monitorze monochromatycznym produkcji Atari.

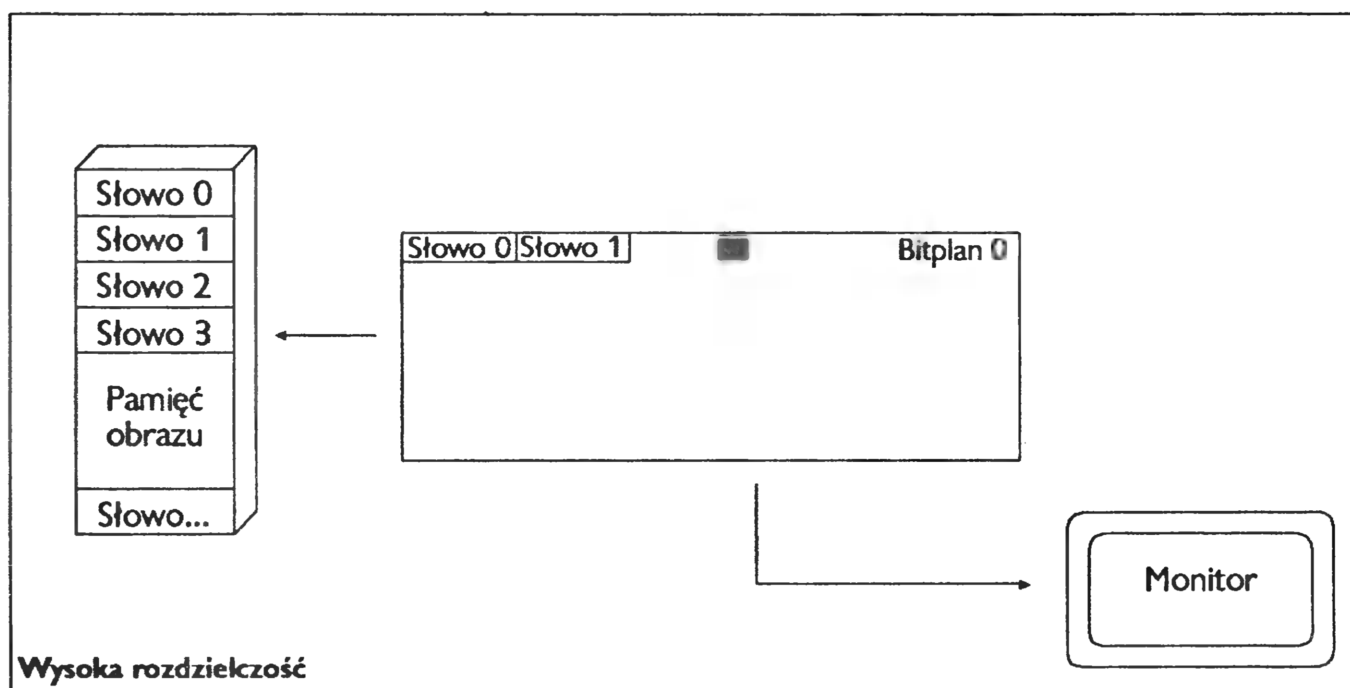
O wyborze rozdzielczości decyduje zawartość rejestru shiftmd:

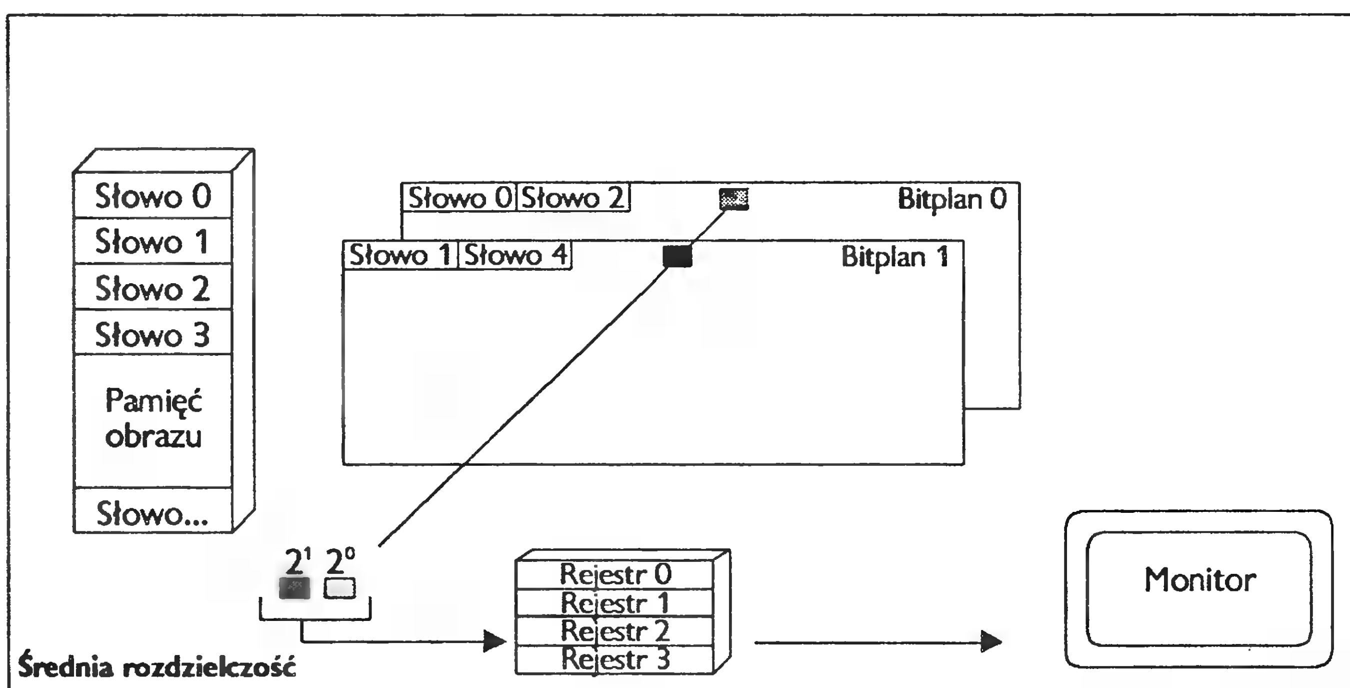
*shiftmd $FFFF8260 (bajt)* — zawartość tego rejestru określa tryb graficzny.

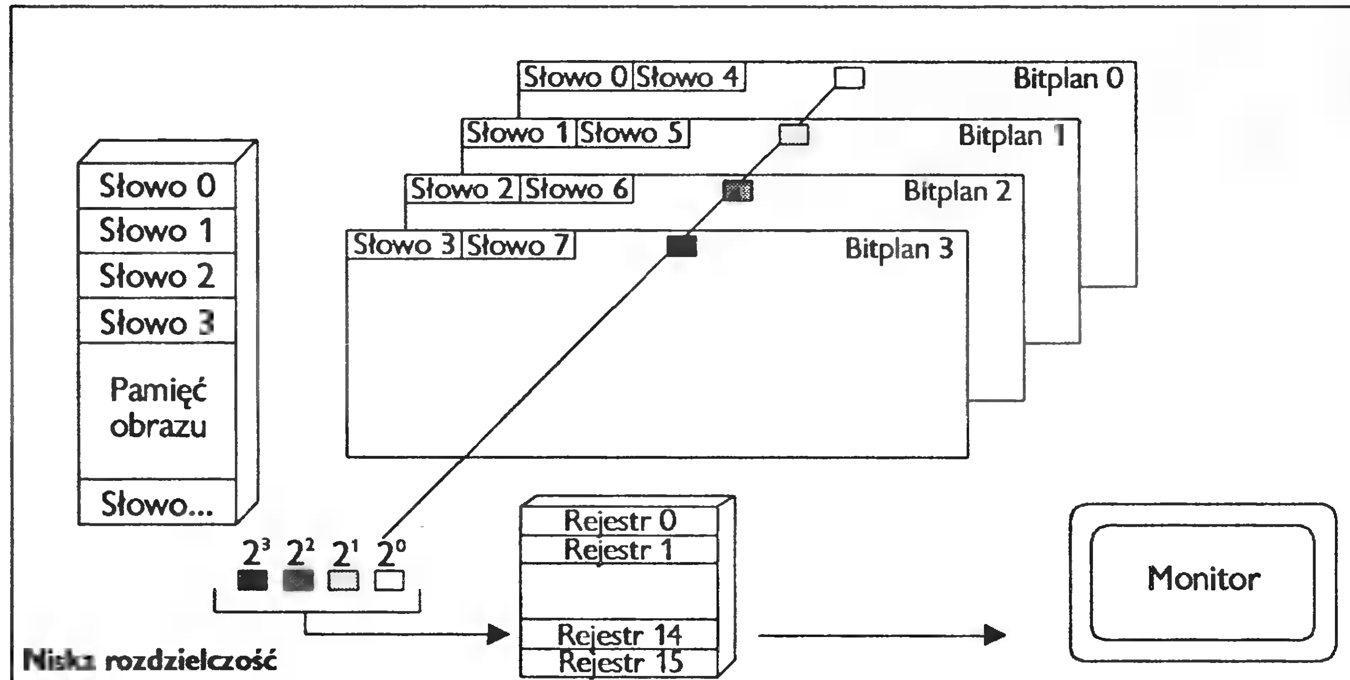
Wartość: 0 — niska rozdzielczość,
1 — średnia rozdzielczość,
2 — wysoka rozdzielczość.




*rys. 1*




*rys. 3 — organizacja pamięci ekranu w różnych rozdzielczościach*

Ponadto istnieje możliwość wyboru częstotliwości ramki dla monitora kolorowego, do czego służy rejestr syncmode:

*syncmode $FFFF820A (bajt)* — zawartość tego rejestru określa tryb synchronizacji:

Bit 0 — ustawiony (1) — synchronizacja zewnętrzna, skasowany (0) — synchronizacja wewnętrzna;

Bit 1 — ustawiony (1) — częstotliwość ramki 50 Hz, skasowany (0) — częstotliwość ramki 60 Hz.

Wszystkie tryby graficzne wymagają 32 000 bajtów pamięci ekranu. Jej organizacja jest następująca:

— rozdzielczość wysoka — w tym przypadku linię tworzy 40 słów (80 bajtów). Kolor każdego punktu jest określony przez odpowiadający mu bit w słowie pamięci, więc mamy do dyspozycji dwa kolory (czarny i biały).

— rozdzielczość średnia — w tym trybie kolor punktu (a właściwie numer rejestru koloru) jest określony przez odpowiednie bity dwóch kolejnych słów (a więc linię tworzy 80 słów — 160 bajtów). W trybie tym mamy dwa bitplany, wszystkie parzyste słowa tworzą jeden bitplan, a nieparzyste drugi.

— rozdzielczość niska — tutaj numer rejestru koloru określają odpowiednie bity w czterech sąsiednich słowach, a zatem mamy cztery bitplany. Linia ma długość 80 słów.

Dokładniej strukturę pamięci ekranu można zrozumieć analizując rysunek 2.

Sposób ustalania adresu pamięci ekranu różni się nieco w komputerach serii ST i ST$^E$.

W Atari ST wystarczą do tego 2 rejestry, gdyż przyjęto adresowanie 24-bitowe, a najmłodszy bajt adresu pamięci ekranu jest ignorowany. W związku z tym pamięć obrazu musi się zaczynać od adresu podzielnego przez 256 i umieszczonego w komórkach:

*dbaseh $FFFF8201 (bajt)* — starszy bajt adresu ekranu,

*dbasel $FFFF8203 (bajt)* — średnie bajt adresu ekranu.

Najmłodszy bajt adresu ekranu jest przyjmowany jako zero, natomiast najstarszy bajt nieużywany (adresowanie 24-bitowe).

Zawartość tych komórek może być zarówno odczytywana (w celu ustalenia aktualnego adresu ekranu) jak i zapisywana (w celu zmiany adresu pamięci ekranu).

W komputerach serii ST$^E$ sprawę ustalania adresu pamięci ekranu rozwiązano nieco lepiej. Adres pamięci ekranu (również 24-bitowy) nie musi być już podzielny przez 256. Dane przeznaczone do wyświetlania muszą być umieszczone od adresu parzystego zawartego w rejestrach:

*dbaseh $FFFF8201 (bajt)* — starszy bajt adresu ekranu,

*dbasel $FFFF8203 (bajt)* — średni bajt adresu ekranu,

*dbaselow $FFFF820C (bajt)* — najmłodszy bajt adresu ekranu.

Jak widać, dołożenie nowego rejestru wykonano w sposób zapewniający maksymalną zgodność. Rejestry te, podobnie jak ich odpowiedniki w ST$^E$ można zarówno odczytywać, jak i zapisywać. Istnienie dodatkowego rejestru dbaselow pozwala na uzyskanie w komputerach serii ST$^E$ płynnego przesuwu w pionie jedynie poprzez modyfikacje tych rejestrów, a także pozwala na swobodniejsze zmiany położenia pamięci obrazu.

Wymienione powyżej rejestry (zarówno w ST$^E$, jak i ST) charakteryzują się tym, że jakakolwiek zmiana adresu jest widoczna dopiero po przerwaniu VBL. Dzieje się tak dlatego, że zawartość rejestrów dbase po każdym przerwaniu VBL jest przepisywana (automatycznie, sprzętowo) do rejestrów licznika, które służą do adresowania pamięci. Zawartość rejestrów licznika jest zwiększana po pobraniu każdego słowa danych przeznaczonych do wyświetlenia i przesłaniu go do SHIFTERA. Tak więc licznik ten zawiera adres aktualnie wyświetlanego punktu. Rejestry licznika są dostępne dla programisty pod następującymi adresami:

*vcounthi $FFFF8205 (bajt)* — rejestr licznika (starszy bajt adresu),

*vcountmid $FFFF8207 (bajt)* — rejestr licznika (średni bajt adresu),

*vcountlow $FFFF8209 (bajt)* — rejestr licznika (młodszy bajt adresu).

Niestety w komputerach serii ST rejestry licznika można jedynie odczytywać. Mimo to rejestry te są bardzo pomocne, gdy chcemy ustalić, który punkt ekranu, w danym momencie, jest wyświetlany czy też zsynchronizować przebieg programu z wyświetlaniem.

Znacznie większe możliwości w tym względzie mają komputery serii ST$^E$. Rejestry licznika można w nich nie tylko odczytywać ale i modyfikować. Pozwala to np. zmienić adres, czy uzyskać płynny przesuw fragmentu ekranu modyfikując jedynie trzy rejestry.

Skoro jesteśmy już przy ST$^E$ trzeba wspomnieć, że komputer ten posiada również inne udogodnienie. Istnieje możliwość sprzętowego płynnego przesuwu obrazu w poziomie. Służą do tego dwa dodatkowe rejestry:

*hscroll $FFFF8265 (bajt)* — rejestr przesuwu poziomego,

*linewid $FFFF820F (bajt)* — długość linii logicznej.

Zanim szczegółowo zajmiemy się tymi rejestrami, zastanówmy się nad problemem przesuwania ekranu w poziomie.

Przyjmijmy dla uproszczenia, że pojedynczy punkt opisuje dokładnie jeden bajt (nie jest to prawdą), zaś obraz ma wymiary 160 x 100 punktów, oraz każda linia zajmuje 160 bajtów. Oznacza to, że aby otrzymać adres następnej linii, należy do adresu poprzedniej linii dodać 160. Zatem przyjmując, że pierwsza linia ekranu ma adres 0, druga linia zaczyna się od adresu 160, trzecia 320 itd. Jeżeli w tym momencie będziemy chcieli przesunąć obraz o 1 punkt w prawo, to pierwsza linia zacznie się od adresu 1, druga 161, trzecia 321. W miejscu gdzie był początek linii drugiej nagle pojawi się koniec pierwszej! W tym momencie łatwo sobie wyobrazić, że linie pionowe rozsypią się nam na szereg punktów i obraz straci swoją organizację. Aby tego uniknąć należy przyjąć, że każda linia ma długość np.: 180 bajtów i tyle też trzeba dodać aby uzyskać adres następnej linii, natomiast wyświetlane jest tylko 160 bajtów. Teraz mając „zapas" 20 bajtów możemy spokojnie przesuwać widoczne 160 punktów. W taki właśnie sposób został rozwiązany sprzętowy przesuw ekranu. Liczba bajtów wyświetlana na ekranie (160 w naszym przykładzie) to długość linii fizycznej, zaś liczba bajtów, którą należy dodać aby uzyskać adres kolejnej linii (180 w naszym przykładzie) to długość linii logicznej.



W normalnych warunkach w Atari ST^E (podobnie jak w ST) linia fizyczna i logiczna mają tą samą długość 160 bajtów w rozdzielczości niskiej i średniej lub 80 bajtów w wysokiej. Aby skorzystać z płynnego przesuwu należy wydłużyć linię logiczną (długości linii fizycznej nie można prosto modyfikować). W tym celu należy wpisać do rejestru linewid o ile słów jest dłuższa linia linia logiczna niż fizyczna. Następnie do rejestru hscroll należy wpisać o ile bitów mają zostać przesunięte wyświetlane linie fizyczne względem linii logicznej (znaczące są jedynie 4 najmłodsze bity, co pozwala na maksymalne przesunięcie o 15 bitów). Dokładniej wyjaśni to rysunek 3.

Teraz pozostało się zająć rejestrami kolorów. Jest ich 16 i każdy z nich ma długość słowa. W Atari ST dla każdej składowej RGB przyporządkowano 3 bity co daje 512 kolorów, zaś posiadacze ST^E mają do dyspozycji 4 bity i 4096 kolorów. Rejestry kolorów umieszczone zostały następująco:

*color0 $FFFF8240 (słowo)* — rejestr koloru 0,
*color1 $FFFF8242 (słowo)* — rejestr koloru 1,
*color2 $FFFF8244 (słowo)* — rejestr koloru 2,
*color3 $FFFF8246 (słowo)* — rejestr koloru 3,
*color4 $FFFF8248 (słowo)* — rejestr koloru 4,
*color5 $FFFF824A (słowo)* — rejestr koloru 5,
*color6 $FFFF824C (słowo)* — rejestr koloru 6,
*color7 $FFFF824E (słowo)* — rejestr koloru 7,
*color8 $FFFF8250 (słowo)* — rejestr koloru 8,
*color9 $FFFF8252 (słowo)* — rejestr koloru 9,
*color10 $FFFF8254 (słowo)* — rejestr koloru 10,
*color11 $FFFF8256 (słowo)* — rejestr koloru 11,
*color12 $FFFF8258 (słowo)* — rejestr koloru 12,
*color13 $FFFF825A (słowo)* — rejestr koloru 13,
*color14 $FFFF825C (słowo)* — rejestr koloru 14,
*color15 $FFFF825E (słowo)* — rejestr koloru 15.

Każdy rejestr koloru ma następującą organizację:

```
bit 15................0
    xxxxrRRRgGGGbBBB
```

gdzie: R, G, B — bity odpowiadające wartościom 0–7 składowych R, G, B w Atari ST,

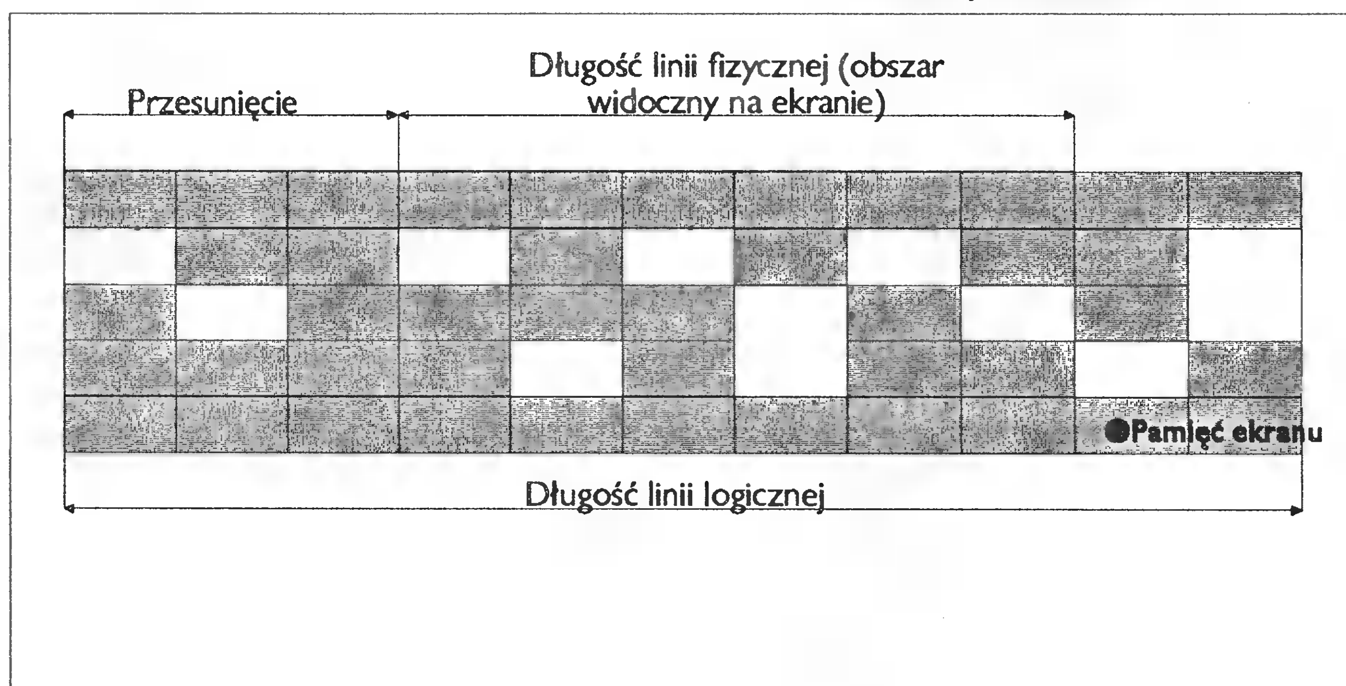
r, g, b — dodatkowe bity, rozszerzające paletę w ST^E,

I to już wszystko, co można napisać odnośnie grafiki. Mimo iż możliwości sprzętu w tym zakresie dzisiaj nie wyglądają rewelacyjnie, to należy pamiętać, że wiele zależy od programu. Jak pokazuje praktyka, programiści mają nieograniczoną fantazje i nie znają słowa niemożliwe, co pozwala im wyświetlać obrazki w rozdzielczości 416 x 276 (overscan), czy też w 32000 kolorów (Photochrome). Jeżeli i Ty chcesz wiedzieć jak to zrobić — czytaj ST^Efana!

Literatura:

[1] ST, STE, TT Profibuch (Jankowski, Rabich, Reschke — Sybex Verlag).

* *co do wymiarów ekranu i ramki występują nieścisłości w dostępnej dokumentacji.* ❑




*rys. 3*

# Zasady prenumeraty

**W poprzednim numerze ostrzegaliśmy, że po wprowadzeniu VAT-u mogą nastąpić zmiany w cenie prenumeraty. Szczęśliwie udało nam się utrzymać dotychczasową cenę pisma, jednak wzrosła cena dyskietek do ST^Efana. Obecnie dyskietka kosztuje nie 15 lecz 20 tys. zł. Oczywiście dotychczasowi prenumeratorzy nie odczuwają żadnych bolesnych zmian...**

**Prenumeratę można zamówić od dowolnego numeru, nawet archiwalnego — dysponujemy jeszcze numerami 1–2/93 oraz 3–4/93. Prenumerata obejmuje sześć miesięcy, czyli trzy numery dwumiesięcznika (lub sześć miesięcznych, jeśli zmieni się cykl wydawniczy) i kosztuje 210 000 zł wraz w dyskietkami. Dodatkowo na prośbę Czytelników wprowadzamy możliwość zaprenumerowania „ST^Efana" bez dyskietek. Półroczna prenumerata kosztuje wtedy 90 tysięcy złotych.**

**Należy zwrócić uwagę na to, że zamieszczony we wkładce kupon zamówieniowy nie jest już aktualny — wydrukowany on został już kilka miesięcy temu, jest to już ostatnia partia. Dlatego prosimy w CZYTELNY sposób zaznaczyć, czy chodzi o prenumeratę z dyskietkami (210 000 zł) czy też bez (90 000 zł)!!!**

**Aby otrzymywać „ST^Efana i dyskietkę" pocztą wystarczy:**

1) Wypełnić zamieszczony na karcie zamówieniowej kupon prenumeraty, nie zapominając o pełnym i wyraźnym adresie zamawiającego, nie zapominając także o zmianach w tym kuponie.

2) Wybrać i zaznaczyć na kuponie formę płatności,

3) Wpłacić na konto firmy Atar system pieniądze i dołączyć do listu kopię wpłaty — w przypadku płatności na konto, lub wysłać po prostu kupon — gdy wybrane zostanie zaliczenie pocztowe. W tym ostatnim przypadku do pobrania doliczone zostaną związane z tym koszty, około 20 tys. zł. Zaliczenie opiewać więc będzie na 230 000 zł lub odpowiednio 110 000 zł.

Kupon prenumeraty pozwala także na zamówienie numerów archiwalnych czasopisma. Nakład pierwszych trzech wydanych numerów (1/92, 2/92, 3/92) definitywnie się wyczerpał, jednakże dla upartych kolekcjonerów przygotowaliśmy pewną ilość kserokopii, które są jeszcze dostępne, w cenie 20 tysięcy za sztukę. Egzemplarze z tego roku (1–2/93, 3–4/93) są normalnie osiągalne.

Nie ukrywamy, że w przypadku prenumerat mile widziane są wpłaty na konto, gdyż zaliczenie pocztowe stanowi ogromną pracę biurokratyczną związaną z pocztą. Poza tym zaliczenie pocztowe powoduje dodatkowe koszty. Warto zwrócić uwagę, że inne czasopisma nie umożliwiają tej formy wysyłki. Jednakże by pozostawić Czytelnikom możliwość wyboru honorujemy oczywiście życzenie zamawiającego. ❑

# GO TO WORK WITH A SMILE!




## GO WITH ...

Notator Logic

OBJECT ORIENTED MIDI RECORDING
AND NOTATION SYSTEM

INTERNATIONALE
MUSIK MESSE
FRANKFURT
STAND-NR.: 9.0 E90

EMAGIC
...we make computers groove.

Atar System
ul.Trzemeska 12, p.412, 53-679 Wrocław
tel/fax (071) 556460

# MIDISTerstwo

## czyli klub użytkowników komputerów Atari i MIDI

*Są ludzie, którym słowo Atari kojarzy się z szarym, żeberkowanym urządzeniem stojącym obok syntezatora (lub kilku). Owo urządzenie służy takim ludziom do uruchamiania ulubionego narzędzia pracy: Cubase'a, Notatora lub innego podobnego. Właśnie tym ludziom – muzykom – dedykujemy nowy klub, który znajdzie swoje miejsce w „ST$^{E}$fanie".*

*Redakcja*

Witajcie wszyscy ci, którzy zajmują się muzyką. Oto powstał nowy dział — MIDISTerstwo, specjalnie dla Was. Nie może się przecież zdarzyć, by w magazynie poświęconym komputerom Atari zabrakło działu, w którym Atari ma najwięcej do powiedzenia. Słuchając wielokrotnie audycji radiowych, puszczając sobie kasety znanych zespołów nie zdajemy sobie sprawy, że przy tworzeniu muzyki wykorzystywano właśnie komputery Atari. Oglądając telewizyjne reklamy możemy posłuchać, jak pracuje Cubase, Notator czy inny program sekwencyjny. Pod względem ilości i jakości oprogramowania MIDI, z Atari może konkurować jedynie Mac. Najczęściej w studio spotkamy duet: Atari + Mac. Firma Atari wypuszczając na rynek Falcona ma szansę ponownie podbić rynek muzyczny, jak kiedyś zrobiła to modelem ST. Zarzuty ze strony różnych amigowców, czy pecetologów, którzy widzą w ST tylko „A i Greka" i myślą, że ze swymi nędznymi 8 bitami dźwięku mają na szerokim polu profesjonalnej muzyki coś do powiedzenia, są bezpodstawne. Wystarczy wybrać się na targi muzyczne do Frankfurtu, które reprezentują najnowsze i najlepsze osiągnięcia w tej właśnie dziedzinie sztuki — można stwierdzić ogromną popularność Atari wśród muzyków.

Dlatego postanowiliśmy utworzyć w ST$^{E}$fanie MIDISTerstwo. Jednocześnie apelujemy:

**Jeżeli jesteś muzykiem, posiadasz komputer Atari, tworzysz dowolny rodzaj muzyki od techno, poprzez rock i electronic do death metalu, napisz do MIDISTerstwa.**

A naprawdę będzie się to opłacało. Planujemy bowiem stworzyć bank programów MIDI Public Domain, bank utworów muzycznych w formacie MIDI-File, bank brzmień do najpopularniejszych instrumentów elektronicznych i... to zależy od Was, drodzy muzycy. Napiszcie, co byście chcieli znaleźć w MIDISTerstwie. Z naszej strony proponujemy Wam jeszcze:

1. Możliwość zaprezentowania się na łamach MIDISTerstwa. Prosimy o informacje kim jesteście, jaki rodzaj muzyki tworzycie, jakich instrumentów używacie i z jakiego programu korzystacie (oczywiście na Atari).

2. Lista przebojów — czyli najlepsze hity stworzone na ST. Każdy może nadesłać swoje propozycje, jak również swoje własne utwory na kasecie. Jeżeli ta forma prezentacji spotka się z szerokim oddźwiękiem, może w przyszłości wydamy kasetę zatytułowaną „The best of Atari ST and MIDI" wzorem firm Atari i Roland, które wydały już dawno odpowiedni kompakt.

3. Wywiady z różnymi ciekawymi muzykami i ich opinie na temat programów na Atari.

4. Opisy sprzętu i oprogramowania.

5. STudio, czyli swoiste konsylium związane z MIDI i Atari.

6. MIDISTyczny Hyde Park m.in. „co to są trackery..."

7. Skrzynka kontaktów i adresów. Firmy muzyczne i software'owe, muzycy, sklepy muzyczne, itp.

8. Oraz wiele innych rzeczy, które mogą znaleźć miejsce w MIDISTerstwie w zależności od Waszych listów.

W związku z pytaniami, jakie się mogą niektórym nasunąć, piszemy:

**Do naszego klubu MIDI mogą należeć wszyscy, którzy posiadają Atari i zajmują się muzyką.**

**Nie akceptujemy tzw. modułów muzycznych, znanych z programów typu tracker. Swe dzieła można nadsyłać albo na kasecie, albo w formacie MIDI-File, ewentualnie w formatach Cubase lub Notator.**

**Do każdego utworu musi być dołączony opis — jaki instrument był użyty oraz opis przyporządkowania brzmień kanałom MIDI.**

**Jako że różni muzycy używają różnych instrumentów, w MIDI-File nie powinny się znajdować SysEx'y do konkretnych typów. Preferowany format General MIDI.**

**W razie niemożności nadesłania nam swego utworu na dysku, prosimy o nadesłanie go na kasecie. Trzeba jednak zaznaczyć z jakich instrumentów i programów się korzystało.**

**W przyszłości przewidziane są konkursy. Wszystko jednak w odpowiednim czasie.**

Tak więc MIDISTerstwo zostaje oficjalnie otwarte. Jeszcze raz zapraszamy wszystkich muzyków. Nie ma się czego wstydzić. Niecierpliwie czekamy na listy zainteresowanych tym działem osób. W zależności od ilości tych listów, MIDISTerstwo będzie się rozwijało. Patronat nad MIDISTerstwem sprawować będzie **Śląskie Stowarzyszenie Muzyki Elektronicznej „Laserion"** z siedzibą w Katowicach i oddziałem we Wrocławiu. Osobą odpowiedzialną za opracowanie działu MIDISTerstwa będzie Martin MIDIan McKey. Wszelkie listy prosimy kierować na adres redakcji ST$^{E}$fana z dopiskiem **„MIDISTerstwo"**.

A teraz radzę wszystkim włączyć radio (program III PR, sobota, godz. $22^{05}$ oraz czwartek, godz. $14^{50}$) i wysłuchać najbardziej atarystycznych audycji w PRiTV czyli „Studio El-Muzyki" oraz „Top Tlen"...

Z muzycznym pozdrowieniem,

Martin MIDIan McKey.

P.S. Niektórym może się wydawać, że preferujemy muzykę elektroniczną. NIE. Każdy muzyk używający Atari będzie w MIDISTerstwie mile widziany (nawet death metalowiec).

❑

# Nowości NEWS MIDI

**Korg** wypuścił nowy instrument: Interactive Music Workstation3. Więcej informacji niedługo.

**Musik-Media**, wydawnictwo niemieckie („Keyboards", „Sticks", „Gitarre & bass", „Production Partner") oferuje książkę zatytułowaną „Synthesizer von gestern", czyli „Syntezatory z wczoraj". Matthias Becker i Klaus Stühlen dołączyli do niej 2 płyty CD z prezentacją możliwości starych syntków. Wyposażyli je w interfejs MIDI i podłączyli do komputera Atari ST z programem Notator. Więcej informacji znajdziecie w czasopiśmie „Keyboards".

**Wszyscy** z niecierpliwością czekają na wersję **Audio** programu **Notator Logic** na Falcona. Firma Emagic zapowiada, że ma się ukazać już wkrótce.

**Akai** prezentuje nową rodzinę samplerów — S3200, S2800... do każdego z tych samplerów sprzedawany jest dodatkowo edytor na Atari ST.

**Firma** E-Mu Systems również dołącza do sprzedanych instrumentów edytory na ST lub na Maca.

**Ostatnio** w audycji „Studio El-muzyki" Jerzy Kordowicz prezentował twórczość jeszcze jednego słynnego STkowca z Holandii — Edvina van der Laaga. Ostatnio słuchaliśmy fragmentów płyty „Syntech Atmospheres". W ogóle warto słuchać tej audycji — dużo w niej o Atari!!!

**Być może** nie wszyscy wiedzą, że istnieje międzynarodowa sieć komputerowa PAN Network przeznaczona specjalnie do muzyków. Należy do niej większość poważniejszych muzycznych potentatów — m.in. Atari Corp.

**Istnieje** coraz to więcej różnych programów muzycznych na Falcona. Korzystając z nieograniczonych możliwości, jakie daje DSP, a ten w Falconie jest jednym z najpopularniejszych w muzyce, możemy liczyć na to, że Atari nadal znajdować się będzie w czołówce firm muzycznych. Czekamy na **Avalon for Falcon**, **Band in a Box DSP** i wiele innych, ciekawych, niesamowitych...

**Wszyscy** narzekają na obecną sytuację Atari (najbardziej jednak w Polsce). Tymczasem np. w Holandii wychodzą płyty CD, które firmują się właśnie przez znaczek Fuji.

**Na rynku** pojawia się coraz więcej przystawek do Falcona, które umożliwiają synchronizację DSP i Codeca z odtwarzaczami CD i magnetofonami DAT. Co ambitniejsze posiadają dodatkowo kolejne układy Codec, lub 24-bitowe przetworniki (po co „marnować" niewykorzystane 8 bitów DSP?).

**Firma Hybrid Arts** planuje wypuszczenie nowego systemu **ADAP**, który w poprzednich wersjach dostępny był na ST i TT. Obecnie możliwe jest to, że pojawi się nowy system, który oprócz DSP Falcona zawierał będzie kilka nowszych DSP Motoroli — 96002, które określane są jako 96 bitowe... oczywiście wszystkie poprzednie funkcje będą zachowane.

**Licencjonowani** posiadacze programów dawnej firmy C-LAB, obecnie Emagic, czyli użytkownicy Notatora, Creatora proszeni są o kontakt z firmą Atar System, dystrybutorem Emagica. Prowadzona jest akcja update'owa programów, związana z pojawieniem się Notatora Logic i nowego Atari Falcona.

MIDI *Martin MIDIan McKey*

# Muzyka obrazu i słowa

W naszym klubie MIDI będziemy rozmawiać z osobami, które w swej pracy wykorzystują komputer Atari ST. MIDISTer czeka na odzew innych muzyków, o których być może nie słyszał, a którzy chcieliby się zaprezentować w MIDISTerstwie. Martin MIDIan McKey przedstawia nam księdza Korneliusza Matauszka, który mieszka i pracuje w Bytomiu. Ksiądz Kornel tworzy muzykę elektroniczną, ilustracyjną. Jako motto przytoczymy tu słowa, które znajdują się na kasecie „Życie":

**„Muzyka wyrywa z chaosu dźwięków**
**dotykając serca**
**Obraz nie zwykł mówić tylko płótnem**
**— odkrywa głębię**
**Słowo wyznacza sens**
**Tworząc pomost**
**pomiędzy Istotą a Istnieniem."**

**Martin McKey: Jak to się wszystko zaczęło, proszę księdza?**

**X.Korneliusz:** Moje spotkanie z muzyką elektroniczną zaczęło się już dość dawno, jednak pierwszy syntezator, **Roland D50**, kupiłem w firmie Langowski w 1987 roku. Przez rok czasu uczyłem się gry na tym instrumencie oraz programowania brzmień.

**M.MK: Jak wiem D50 był jednobarwowy. Czy już wtedy grał ksiądz jakieś koncerty?**

**K.M.:** Miałem swoje występy na różnych rekolekcjach, czy dniach skupienia. Pierwszy większy koncert to Bliskie Spotkania z Muzyką Science Fiction w Warszawie w 1989 r.

**M.MK: I zaczęło brakować księdzu sekcji akompaniującej?**

**K.M.:** Można i tak to nazwać. Na **D50** mogłem grać na żywo na klawiaturze jedną barwą.

**M.MK: I co było dalej?**

**K.M.:** Kupiłem sobie moduł **Korga M3R**. Niestety ani on, ani **D50** nie posiadały sekwencera. Tak więc nie mogłem w pełni wykorzystać tzw. multitimbral mode w **Korgu M3R**. Do tego potrzebny by był sekwencer.

**M.MK: Tak, wiem. I wtedy kupił ksiądz komputer.**

**K.M.:** Tak. Wybrałem właśnie komputer **Atari ST**, gdyż posiada on najlepsze oprogramowanie na MIDI.

**M.MK: Na jaki program się ksiądz zdecydował?**

**K.M.:** Wybrałem program **Notator** firmy C-Lab *[obecnie Emagic — przyp. red.]*. Jest to rewelacyjny program i z całą pewnością mogę stwierdzić, że w pełni zaspokaja moje potrzeby w trakcie komponowania. Posiada on również możliwość wydruku partytury z bardzo dobrą jakością. Uważam go za dość prosty w obsłudze.

**M.MK: Słyszał ksiądz być może o konkurencyjnym programie firmy Steinberg, czyli Cubase. Co ksiądz o tym sądzi? Czy lepszy Notator czy Cubase?**

**K.M.:** Trudno powiedzieć. Ja uważam, że lepiej pracuje mi się z **Notatorem**. Jest to chyba jednak kwestia przyzwyczajenia. Trudno porównywać tu te programy. Wybór nie jest łatwy. **Notatora** używam od wersji 2.2, obecnie posiadam wersję 3.1. Przyzwyczaiłem się już do **Notatora** i nie zamieniłbym go teraz na **Cubase**.

**M.MK: Powróćmy jeszcze do koncertów. Kiedy odbył się taki pierwszy poważny koncert, na którym ksiądz wystąpił?**

**K.M.:** Było to... w roku 1989, w lutym. Koncert nazywał się „Bliskie Spotkania z muzyką Science Fiction" i odbył się w klubie Riwiera Remont. W 1990 grałem po raz drugi na „Bliskich spotkaniach ...", ale już z całym sprzętem, włącznie z komputerem **Atari**.

**M.MK: A następne?**

**K.M.:** Jak już wspomniałem, grywałem na różnych rekolekcjach czy dniach skupienia. Drugim moim większym koncertem był „Zlot Elektronicznych Fanatyków" ZEF'90 w Karpaczu. A potem grałem w Planetarium Śląskim, na koncercie „Spotkanie z muzyką gwiazd"...

**M.MK: ...pamiętam, sami go organizowaliśmy...**

**K.M.:** ...no właśnie. Z Józefem Skrzekiem. Następnie miałem swój występ na Górze Szybowcowej w 1992 r. Wprawdzie nie zagrałem tam zbyt długo, ale w niezłym towarzystwie.

**M.MK: No tak. Marek Biliński, Grzegorz Stróżniak, Walk Away, Marek Pluciński... Porozmawiajmy teraz o ostatniej kasecie księdza pt. „Życie".**

**K.M.:** „Życie" nie było moją pierwszą kasetą. Wcześniej nagrałem kasetę podczas mego występu na ZEF'90, którą nagrał Ziemowit Poniatowski i która nosiła tytuł „6 nad ranem". Natomiast „Życie" jest moją pierwszą kasetą, którą sam firmowałem. Praca nad nią trwała około półtora roku. Utwory na tej kasecie powstawały w latach 1991 i 1992. Inspiracje muzyczne czerpałem z Biblii. Jest to muzyka elektroniczna, ilustracyjna. Skłaniająca do refleksji. Na pewno nie nadaje się do słuchania w dyskotekach.

**M.MK: A najbliższe plany na przyszłość? Może jakaś nowa kaseta?**

**K.M.:** Tak. Planuję wydać drugą kasetę, która na razie nosi roboczy tytuł „Życie II". Ale może wymyślę jakiś inny. Złożą się na nią utwory nowe, które ciągle tworzę. Będą to kompozycje z lat 1992 i 1993. Nie mogę jeszcze dokładnie powiedzieć, kiedy będzie dostępna.

**M.MK: A teraz coś dla Czytelników ST$^E$fana...**

**K.M.:** Bardzo się cieszę, że jest taka gazeta. Prowadzona jest ciekawie. Uważam **Atari ST** za bardzo dobry komputer. Jego oprogramowanie na MIDI jest naprawdę rewelacyjne. Oprócz **Notato ra**używam jeszcze **Explorera M1** do mojego modułu **M3R**. Pamięć w komputerze rozbudowałem do 2,5 MB. Naprawdę, klub MIDI powinien mieć miejsce w ST$^E$fanie. W miarę możliwości służył będę swoimi spostrzeżeniami i chętnie będę współpracował z MIDISTerstwem.

**M.MK: Niezmiernie mi miło. Ostatnio nasila się tendencja w muzyce nazywana „Unplugged". Czy elektronicy mają w tej sprawie coś do powiedzenia?**

**K.M.:** Ten cały „Unplugged" to w gruncie rzeczy „hiperplugged". Na scenie wygląda to tak akustycznie, gitary, fortepiany. Tymczasem za kulisami aż roi się od różnych kamer pogłosowych, systemów nagrywających, wzmacniaczy, transmiterów, nagłośnienia. „Unplugged" moim zdaniem spowodowany jest właśnie przez elektronikę. Obecnie istnieje tyle różnych instrumentów, systemów, wzmacniaczy i innych urządzeń, że tworzy się pewien przesyt. Dlatego niektórzy szukają szansy, którą widzą w instrumentach akustycznych. Tymczasem taki koncert „unplugged" w rzeczywistości wygląda nie tak bardzo antyelektronicznie...

**M.MK: A więc wszędobylska elektronika. Słyszałem ostatnio księdza w radiu katolickim „Puls". Prowadzi tam ksiądz swoje audycje?**

**K.M.:** Tak, mam tam swoją audycję. Puszczam tam muzykę elektroniczną i staram się ilustrować nią cytaty z Biblii. Dołączam też własne przemyślenia.

**M.MK: Bardzo dziękuję za rozmowę. Myślę, że pozostaniemy w kontakcie. W naszym klubie MIDI będziemy się starali nie tylko przedstawiać ciekawych ludzi, ale i służyć radą.**

**K.M.:** Bardzo dobrze. Interesuje mnie wasza propozycja stworzenia banku programów PD na MIDI oraz banku MIDI File. Można nawet poszerzyć zbiór programów o dyskietki z brzmieniami do konkretnych instrumentów. Chętnie wymieniłbym się brzmieniami **D50** oraz **M3R**. Dotychczas w Polsce nie było poważnej gazety dla użytkowników MIDI. Wprawdzie ostatnio pojawiają się różne „Obszary dźwięku" czy taki „Muzyk", jednak brakuje gazety stricte dla elektroników, takiej jak choćby niemiecki „Keyboards". O, przepraszam. Był kiedyś „Moogazyn", wydawany w Warszawie i dostępny jedynie w prenumeracie oraz dostarczany przez pocztę. Szkoda, że zaprzestano jego wydawania. Była to właśnie gazeta dla elektroników. Wydawali ją m.in. wspomniany Ziemowit Poniatowski, Artur Lasoń i inni. ST$^E$fan, z racji tego, że zajmuje się komputerami Atari, prowadząc dział MIDI, ma szansę stać się pierwszym pismem właśnie dla użytkowników elektronicznych klawiatur i modułów. Jestem pewien, że większość polskich muzyków posiada komputery Atari. Mówi się wprawdzie dużo o MacIntoshu, jednak komputery te pozostają poza zasięgiem przeciętnego muzyka. Nie mówiąc już o tym, że potrzebują drogiego oprzyrządowania, interfejsu MIDI i tak dalej. Sądzę, że nawet ci, którzy mają MacIntoshe, nie zrezygnują tak prędko ze swoich starych maszynek. Następna sprawa to Falcon. Uważam, że właśnie Falcon będzie największym konkurentem ST czy TT w dziedzinie muzyki. A wracając do działu MIDI to ze swej strony pomogę w miarę swoich możliwości. Jeżeli ktoś będzie miał problem z **Notatorem**, służę pomocą.

**M.MK: A więc przypomnijmy, obecnie ksiądz pracuje na...**

**K.M.: Atari 1040 ST** z 2,5 MB RAM, **Roland D50** jako klawiatura muzyczna i sterująca, **Korg M3R** jako moduł brzmieniowy w sekcjach sekwencera, którym jest rewelacyjny program **Notator v.3.1**.

**M.MK: Planuje ksiądz w najbliższym czasie rozbudować swoje instrumentarium?**

**K.M.:** Istnieje teraz wiele bardzo dobrych instrumentów. Będąc na targach muzycznych we Frankfurcie nie zauważyłem jednak znaczących nowości. Obecnie zwrócono uwagę na „polepszenie" brzmień starych, sprawdzonych instrumentów. I tak, do **D50** istnieje przystawka multitimbral, do **Korga M1** przystawka **PlusOne**. Ja oczywiście chciałbym rozbudować swoje instrumentarium. Przydałby mi się taki **Korg O3R/W** czy **Wavestation A/D**, które mają naprawdę rewelacyjne brzmienie. Ciekawy instrument zaprezentowała firma **E-Mu**, która pokazała moduł **Vintage Keys**, zawierający brzmienia wszystkich starych dobrych analogów:**Mooga, Oberheima, Yamahy, Mellotronu** i innych. Jeżeli będą pieniądze, to rozbuduję swe instrumentarium.

**M.MK: A co sądzi ksiądz o nowym komputerze Atari, Falconie?**

**K.M.:** Jeżeli tylko muzycy dowiedzą się o jego możliwościach i dostaną odpowiednie programy do swoich potrzeb, to myślę, że zastąpi on nie tylko **ST**, ale i poważniejsze systemy na Macach. Jednak niektórzy zarzucają mu to, że posiada on obudowę 1040-stki. Wielu muzyków widziałoby go w obudowie tower.

**M.MK: No tak, ale kto potem będzie zabierał komputer, klawiaturę, monitor itd. na koncert?**

**K.M.:** No więc zdania w tej kwestii są podzielone.

**M.MK: A czy oprócz muzyki, wykorzystuje ksiądz komputer do innych celów?**

**K.M.:** Ostatnio kupiłem drukarkę atramentową DeskJet 510. Zaopatrzyłem się również w edytor **WordPlus**.

**M.MK: Czekamy wobec tego niecierpliwie na artykuły do klubu MIDI w ST$^E$fanie. Czy zechce ksiądz z nami współpracować? Czy możemy liczyć na jakieś artykuły?**

**K.M.:** Oczywiście. Postaram się z wami współpracować.

**M.MK: No to teraz już naprawdę serdecznie podziękowanie za rozmowę.**

**K.M.:** Z Panem Bogiem.

Rozmowę przeprowadził w sierpniu 1993 Martin MIDIan McKey.

MIDI *Martin MIDIan McKey*

# DNLT

## czyli o module szumnie muzyką zwanym

Zaczęło się banalnie. Najpierw był ZX Spectrum ze swoim popiskującym głośniczkiem. Potem PC/XT też niewiele lepszy. Aż przyszedł Commodore 64. Ach, co to było! Każdy się zachwycał. Trzy kanały syntezatora! Pięknie. Następne było Atari ST, które jako pierwszy w historii komputer wprowadził interfejs MIDI. Jako jego własny generator pracowała Yamaha YM2149. No i fajnie. Wszystkim się podobało. Aż tu nagle... Amiga ze swoimi 4 przetwornikami C/A. Szok! Cyfrowa muzyka porównywana z CD (???). Wprawdzie takie rozwiązanie dawało lepsze rezultaty niż standardowe generatory, jednak wymagało już trochę programowania. W SID'zie, czy a i greku wystarczyło parę poke i już grało. Tymczasem w przetworniku trzeba było wysłać ciąg poków i to odpowiednio szybko. Ale sam hardware nic nie wart. Na ST pojawiło się kilka rzeczy: Pro24, Creator, DX Heaven, KCS Omega i jeszcze parę innych. Na Amigę tymczasem stworzono Soundtrackera. Zachwycano się jego możliwościami — cóż to za wspaniały edytor prawdziwej muzyki. A na targach we Frankfurcie Notator szalał w konkurach z Cubasem, Synthworks z Explorerem... C-LAB ze Steinbergiem. W międzyczasie zespoły Tangerine Dream, Genesis czy inne za pomocą ST pięły się po szczeblach kariery muzycznej. A Soundtracker mruczał w zaciszu domowego fana gier komputerowych. Przyjrzyjmy się teraz owemu trackerowi...

A więc — program muzyczny. Ale czy na pewno? Czy prawdziwy muzyk korzystałby z niego? Mamy 4 ślady oraz 32 instrumenty do dyspozycji. Niby wszystko OK. Szybko jednak okaże się, że możemy wykorzystać zaledwie 4 dźwięki naraz. Jest to jeden akord czterodźwiękowy. I poza tym nic. Możemy digitalizować całe akordy, jednak takie rozwiązanie jest niemile widziane u muzyka. Pragnąłby on bowiem wykorzystać wszystkie niuanse akordu (np. przewroty). A niestety nie jest to możliwe. Przypuśćmy, że muzyk da sobie z tym radę i znajdzie odpowiedni sposób. Potrzebuje teraz odpowiedniej artykulacji, np. gra legato. Nie pozostaje nic innego, jak zużyć 3 ślady. A co z dynamiką? Co to znaczy C10? A C3A? Czy to jest forte, piano czy pianissimo?

Zapis w trackerach przedstawia się muzykom jako niezrozumiały bełkot. Np.: D#2-02C30. Co to jest? Przyjmijmy jednak, że muzyk wie, co to oznacza. Jednak to co jest tak napisane, wcale nie musi odpowiadać nucie Dis w drugiej oktawie o głośności 30 (ciekawe w jakich jednostkach). Wynika to ze sposobu generowania dźwięku. Wystarczy drobna różnica w częstotliwości próbkowania, żeby z Dis zrobiło się E, czy nawet A. Bądź tu potem mądry dobierając akordy! Czas trwania dźwięku jest ograniczony. Istnieje wszakże możliwość zapętlenia go, ale jest to żmudna robota. Niedbale zapętlony sampel będzie stukał, „zjeżdżał", nie będzie współbrzmiał w akordzie, co jest chyba jego najpoważniejszą wadą. Proszę spróbować stworzyć akord D-A-D2. Pierwszy dźwięk jeszcze dobrze nie nabrzmi, kiedy ostatni dźwięk z tej triady się skończy. A dość typowa sekwencja w muzyce, polegająca na graniu tym samym dźwiękiem w różnych oktawach jest na trackerze nie do pomyślenia. Efekty osiągane można by ocenić na jedynkę z plusem. (Ten plus to za to, że w ogóle coś udało się zrobić). Układając „muzykę" na trackerze łamiemy wszelkie zasady harmonii, dynamiki, agogiki i innych fundamentalnych zasad w muzyce. Najlepiej na trackerze wychodzi tzw. „klepanka" perkusyjna z pulsującym techno basem i tandetną melodyjką. Poważnej muzyki pisanej przez duże „M" na trackerze nie da się utworzyć. Co prawda, niektórym ludziom podobają się „wspaniałe moduły",lecz gusty nie podlegają dyskusji. Dla prawdziwego muzyka jednak, niezależnie od różnych tekstów wypisywanych ku czci trackerów, muzyka nie jest rzeczą, którą można pomiatać. Muzyka, pisana przez duże „M".

Sposób grania modułu przez tracker również zadziwi niejednego muzyka. Odtwarzanie sampli — cóż dziwnego? Tak, w przypadku elektronicznych perkusji rzeczywiście nic. Ale w przypadku instrumentów strojonych... sprawa wygląda wprost tragicznie. Nie dość, że czas trwania dźwięku zmienia się wraz z wysokością, to jeszcze traci on swoją jakość, kiedy częstotliwość odgrywania zacznie się trochę więcej różnić od częstotliwości próbkowania (efekt podobny do efektu uzyskiwanego przez zmianę prędkości obrotów talerza z płytą analogową — w tym przypadku jednak pasmo przenoszenia jest wielokrotnie większe). A któż jest zachwycony słuchając hi-hata w module? To szuranie przypomina raczej zabijanie much łapką z gwizdkiem. W profesjonalnych samplerach sprawa wygląda tak, że próbkuje się kilka, a nawet kilkanaście razy dane brzmienie (w skrajnych przypadkach, nawet każdy klawisz — w elektronicznych pianinach). Dopiero wtedy jakość jest dobra. Następna sprawa to „efekty" uzyskiwane w trackerach. Jakieś vibrata, tremola, glissanda, które przypominają wszystko oprócz tego, co przypominać powinny. Ustawiając niektóre parametry otrzymujemy zaskakujące rezultaty. Sztuczne dodawanie pogłosów, przez wielokrotne zagranie dźwięku o coraz mniejszej głośności. Skracanie wybrzmienia. Zmiana wysokości. Wszystko przez zmianę nikomu nic nie mówiących cyferek.

Pierwszy kontakt muzyka z trackerem zaczyna się pytaniem: „Co to jest?". Następuje potem krótka przerwa i stwierdzenie: „I ty to nazywasz muzyką???". Ani zapis, ani jakość, ani możliwości, chyba tylko łatwość tworzenia pospolitej lipy muzycznej (typowej dla klezmerów — patrz ENTER 8/82) i to jeszcze w stylu techno przyczyniła się do tego, że stał on się tak popularny. Pytając muzyków o programy muzyczne, najczęściej usłyszymy „Notator, Cubase, Pro24, Omega...". Nigdy jednak nie usłyszałem „Soundtracker, Noisetracker, Startrekker...". Tak, tak moi drodzy trackerowi modularze. Nie można was nazwać muzykami. Dobremu praktykowi wystarczy parę minut by stworzyć moduł, porywający wszystkich giełdowych bywalców. Ale taki praktyk nie jest wcale muzykiem. Szokujące jest to, ilu mamy obecnie koderów „muzyki" do komputerów. Przyprowadzając takiego „muzyka" do klawiatury syntezatora, stwierdzamy, że najczęściej bawi się on edycją brzmienia, niż gra coś konkretnego. Jeżeli w ogóle wie, gdzie jest C i umie zagrać tonikę C-dur. Tracker wypacza zmysł muzyczny. Idzie bowiem na zbyt wielką łatwiznę. Tymczasem prawdziwa muzyka to nie kawałki w stylu techno, rap i kilku zdigitalizowanych wypowiedzi. Tworząc muzykę trzeba mieć choćby w średnim stopniu opanowaną klawiaturę (struny, smyczek, pałki), mieć blade pojęcie o harmonii, melodyce, dynamice i trzeba po prostu mieć talent. Tymczasem wiele talentów bezpowrotnie niszczy swój zmysł artystyczny korzystając właśnie z trackerów. Uczniom szkół muzycznych, którym zależy na rozwijaniu twórczości artystycznej zdecydowanie odradzam wszelkiego rodzaju trackery. Jest to swoisty narkotyk muzyczny, który początkowo wywołuje euforię, jednak wypacza po mału zmysł artystyczny.

Tymczasem programów typu tracker powstaje coraz więcej. Ich autorzy mają na celu chyba zaśmiecenie muzyki przez tandetną chałę i lipę. Rytmiczne postukiwania i popukiwania z brzdękającym basem i świdrującymi wskutek niskiej częstotliwości próbkowania wyższymi dźwiękami... Troszkę lepsze kawałki powstają gdy digitalizuje się całe sekwencje tzw. rhythm-loop, czy po prostu frazy muzyczne, które składa się w całość. Jednak zapis wygląda w tym przypadku okropnie myląco i w ogóle nie odpowiada rzeczywistości. Typowa sprawa — korzystając z perkusji zużywamy od 2 do 10 różnych sampli, a zarazem dźwięki do wykorzystania. Tymczasem jedno uderzenie perkusji może być różnie artykułowane np. talerz — uderzony w środku, czy przy brzegu, ewentualnie otwarty lub zamknięty. A ilość dźwięków to tylko 32... Pianino w niższych i najwyższych partiach brzmi fatalnie, talerze „sypią się" w trzeciej oktawie... a właśnie, tylko 3 oktawy, jeszcze bez tzw. mydła (najwyższe C). Można by tak jeszcze pisać o wielu innych wadach: a to sposób zapisu na dysk, a to sposób tworzenia tzw. songu, a to brak swobodnej manipulacji tempem (te Fxx nie pasuje do żadnych, przyjętych norm określania tempa, ani w muzyce, ani w beats per second...). Tak więc, szanowny trackerowy muzyku, pora zastanowić się nad tym, czy aby tracker nie spowodował u ciebie pewnego otępienia muzycznego. Myśląc poważnie o muzyce, trzeba zacząć od pospolitego pianina, gitary, ewentualnie instrumentu elektronicznego. Ważne jest przy tym, żeby w miarę możliwości opanować klawiaturę, poznać podstawy harmonii, rytmiki, dynamiki. Nie wystarczy bezmyślnie odgrywać czyichś utworów. Przy odrobinie wprawy i treningu zrobić to może każdy. A muzyka nie polega wcale na tym. Muzyka jest dziedziną sztuki i zarazem kultury.

Niektórzy mogą poczuć się obrażeni po przeczytaniu tego tekstu. Uważali się dotąd za wspaniałych muzyków, zdobywających laury w komputerowym światku. Jeżeli chcą oni dalej zajmować się muzyką, a nie trackerem, radzę odłożyć ten program i wziąć do ręki starą gitarę, zasiąść przy pianinie, wziąć do ręki smyczek i skrzypce... i zacząć poznawać muzykę od podstaw. Pisząc ten artykuł na łamach gazety poświęconej komputerom Atari, gdzie są one w sytuacji niejako upzywilejowanej, gdyż posiadają mnóstwo profesjonalnych Programów Muzycznych z prawdziwego zdarzenia, łatwo będzie skasować jakiegoś marnego trackera i zdobyć sobie jakiś prosty sekwencer, czy też program do nauki muzyki, pragnę równocześnie zachęcić ich użytkowników do przyjaźniejszego spojrzenia na MIDI. Tu naprawdę zaczyna się prawdziwa muzyka. I nie trzeba Cubase czy Notatora, by nauczyć się grać. Tu trzeba pracy. Odpowiednie narzędzia już istnieją.

Swoją drogą ciekawi mnie pewne zachłystywanie się modułami przez użytkowników Atari. Gromadzą oni, podobnie jak amigowcy, stosy dyskietek z nagranymi kawałkami z trackera. Jedyna zaleta, spośród mnóstwa wad, to możliwość odsłuchania muzyki prosto z komputera, bez dodatkowych urządzeń. Niektórzy traktują MIDI jak coś, co wymaga ogromnych pieniędzy na instrumenty, oprzyrządowanie. Tymczasem istnieje wiele tanich (do 10 mln) instrumentów, które z powodzeniem można w MIDI wykorzystać. Rozumiem, że każdy chciałby mieć sprzęt jak najlepszy. Ja też bym chciał. Tymczasem w większości przypadków muzycy w swoim domu używają z reguły jednego instrumentu klawiszowego, jednego modułu i jakiegoś magnetofonu. Wcale nie potrzeba wykorzystywać wszystkich 16 śladów MIDI. Czasami wystarczą 2 kanały. Z tym, że w MIDI nie istnieją ograniczenia, z jakimi spotykamy się w trackerze. Zresztą, nie ma co porównywać trackerów z programami MIDI. A co najśmieszniejsze, ostatnio pojawiły się trackery z MIDI! Ale ubaw! 4 tracki i 4 barwy przy 16 kanałach MIDI!

Pora kończyć już ten tekst. Mam nadzieję, że okaże się on gwoździem do trumny wszystkich programów trackerowych. I naprawdę, atarowcy, chcąc zajmować się muzyką, zajmijcie się na poważnie dwoma gniazdkami DIN w waszych maszynkach, gdzie widnieje obrazek nutki... W imieniu wielu muzyków z prawdziwego zdarzenia,

Martin MIDIan McKey.

P.S. Ufff, pora włączyć AudioSculpture i posłuchać sobie modułu...

A co znaczy skrót DNLT? To proste... Dlaczego Nie Lubię Trackerów...

Redakcja „ST<sup>e</sup>fana" nie bierze odpowiedzialności za następstwa i szkody moralne spowodowane przez ten tekst...

Krzysztof A. Ziembik



# Słów kilka o grafice i animacji

**Cykl, którego tytuł znajduje się nieco wyżej, w zamierzeniach miał się składać z pięciu części (nie wykluczam, że może być ich więcej) w następującej kolejności: tworzenie grafiki, programy graficzne, tworzenie animacji, programy animacyjne, przykład tworzenia animacji. Podział na części teoretyczne i praktyczne miał ułatwić strawienie tematu. Niestety, z mojej winy drugi odcinek ukazał się przed pierwszym. Wszystkich zainteresowanych przepraszam i obiecuję poprawę. A tymczasem przedstawiam drugi (pierwotnie pierwszy) wykład traktujący o sposobach tworzenia grafiki.**

## Co to jest grafika?

Właściwie ogólna definicja grafiki pojawiła się już poprzednim razem, więc przypomnę tylko dla porządku, że za grafikę należy uważać wszystko, co pojawia się na ekranie. Ponieważ mogą to być zarówno kolorowe obrazki jak i nawet zwykły tekst, staje się zrozumiałe, że do tworzenia grafiki stosuje się przeróżne metody i narzędzia. Nikt, komu zależy na oszczędności czasu i pieniędzy, nie będzie przecież pisał tekstu przy pomocy programów graficznych, a rysunki wykonywał z liter (chyba, że o to artyście chodzi). W skrócie mówiąc, minimalizacja kosztów realizacji i uzyskanie jak najlepszych efektów pracy w jak najkrótszym czasie są główną przyczyną stosowania różnych metod tworzenia grafiki. Jest tych sposobów kilka, z których najpopularniejsze omawiam poniżej.

### Skanowanie

Skanowanie to mówiąc najprościej przenoszenie obrazów istniejących w rzeczywistości na ekran komputera przy użyciu odpowiedniego urządzenia zwanego skanerem. Wyróżnia się wiele typów skanerów, ale są cztery podstawowe podziały: ze względu na rodzaj skanowanych przedmiotów (mamy wtedy skanery do płaszczyzn lub brył), zastosowane odwzorowanie barw (czarnobiałe, w odcieniach szarości i kolorowe), rozdzielczość pracy (300 dpi, 600 dpi, itd.) oraz budowa skanera (ręczny lub stacjonarny). Wymaga to pewnych wyjaśnień. Skanery do płaszczyzn są przeznaczone do przeniesienia na ekran powierzchni płaskich, np. kartki papieru, fotografii. Można próbować skanować nimi obiekty trójwymiarowe, ale i tak efektem będzie zwykły dwuwymiarowy obraz. Co innego ze specjalnie skonstruowanymi skanerami do obiektów trójwymiarowych. Wynikiem ich pracy są bryły, które można obejrzeć ze wszystkich stron za pomocą odpowiedniego programu. Skanery do płaszczyzn stosuje się bardzo często, np. przy komputerowym składzie wydawnictw do przenoszenia wszelkich fotografii i rysunków, przy produkcji gier komputerowych (krajobrazy wzięte z rzeczywistości są na ogół lepsze od tych narysowanych). Skanery trójwymiarowe to domena profesjonalistów, głównie ze względu na koszty. Wykorzystuje się je przy produkcji filmów, przy projektowaniu przez inżynierów, architektów, a także w różnych placówkach badawczych (np. podczas badań nad fizyką ruchu).

Jeśli chodzi o barwy, to podział jest prosty. Skanery mogą być czarnobiałe, jeśli dają obraz dwubarwny, pracujące w odcieniach szarości oraz kolorowe. Te pierwsze stosuje się do przenoszenia czarnobiałych rysunków i tekstu, pozostałe do płaszczyzn i brył kolorowych. Rozdzielczość pracy to kolejny parametr mówiący o jakości obrazu, który pojawi się na ekranie. Im jest ona większa tym lepszy obraz, oczywiście pod warunkiem, że monitor (lub inne urządzenie wyjściowe) również będzie umożliwiał osiągnięcie tej rozdzielczości. Obecnie najpopularniejsze są skanery 300 i 600 dpi, czyli takie, które jeden cal przenoszonego obrazu dzielą na 300 i 600 punktów. Ostatni podział dotyczy warunków pracy. Skaner ręczny musi być prowadzony, tak aby ogarnąć całą rzecz, którą się skanuje. Stacjonarny, jak sama nazwa wskazuje stoi w miejscu, a skanowany obiekt albo się wkłada w skaner (skaner dwuwymiarowy), albo obraca wokół własnej osi (skaner trójwymiarowy).

Skanowanie posiada swoje wady i zalety, które chciałbym przedstawić. Po pierwsze jest to całkowicie „odtwórcza" metoda tworzenia grafiki, no chyba, że skanuje się własne prace. Oprócz tego nie da się tą metodą przedstawić wszystkiego, a tylko to, co już istnieje. Poza tym skanowanie wymaga dużych nakładów sprzętowych (dość drogi skaner, w miarę szybki komputer z dużą ilością pamięci, dobry program skanujący). Na szczęście oprócz wad są też zalety. Nie potrzebne są zdolności artystyczne, aby tworzyć grafikę, można wielokrotnie powtarzać skanowanie z innym ustawieniem parametrów, łatwo przenieść obrazy z rzeczywistości.

### Digitalizacja

Digitalizacja polega na zamianie obrazu analogowego na cyfrowy, który może wyświetlić komputer. Ponieważ dotyczy to przenoszenia obrazu z magnetowidu, kamery wideo, aparatu fotograficznego, telewizora, a urządzenia te przedstawiają rzeczy na ogół istniejące w rzeczywistości, to można powiedzieć, że digitalizacja podobnie jak skanowanie jest procesem, który odzwierciedla obrazy rzeczywistości na ekranie komputera. Także w tym przypadku istnieją różne rodzaje digitalizerów, a raczej różne zestawy, mniej lub bardziej rozbudowane, do grafiki czarnobiałej i kolorowej. Zastosowanie tak stworzonej grafiki jest podobne jak w przypadku skanowania, choć można digitalizować nie tylko pojedyncze obrazu, ale także całe fragmenty np. filmu. Po takie metody coraz częściej sięgają autorzy gier, a także zawodowcy do wyrafinowanych filmów reklamowych.

Również wady i zalety digitalizacji są prawie takie same jak skanowania. Wady to duże wymagania sprzętowe (dodatkowy osprzęt w postaci urządzeń do digitalizacji, a także tych, z których będzie pobierany obraz, szybki komputer, dużo pamięci), brak możliwości swobodnego wyboru tematu, minimum zdolności fotograficznych. Podstawową zaletą digitalizacji jest łatwość przenoszenia obrazu z rzeczywistości bez zbytnich nakładów czasu, ale ważne też jest to, że można to dokonywać wielokrotnych prób, nawet po zmianie wielu parametrów (np. oświetlenia, nasycenia barw).

### Rysowanie odręczne

Chyba nikomu nie trzeba tłumaczyć co oznacza rysowanie odręczne. Jest to według mnie podstawowa metoda tworzenia amatorskiej grafiki komputerowej. Nawet jeśli na początku rysunek jest tworzony w inny sposób, to i tak często ostateczne poprawki robi się właśnie ręcznie. Ważne jest też to, że najczęściej za robienie grafiki biorą się ci, którzy mieli już doświadczenie ze zwykłymi przyborami do rysowania (kredka, pędzel czy długopis) i odręczne rysowanie na ekranie jest naturalną formą tworzenia. Jednak największą zaletą, która sprawia, że warto rysować ręcznie, jest to, iż nie ma praktycznie ograniczeń zarówno treści jak i formy, a wszelkie ograniczenia zawierają się w zdolnościach rysującego. Wydaje mi się, że ta metoda będzie stosowana przez atarowców coraz częściej, ponieważ pojawiły się na Falcona wspaniałe programy malarskie, które rysowanie odręczne stawiają w nowym świetle. 65 tysięcy kolorów pozwala na dowolne operowanie światłem i barwą, a dostępne rozdzielczości powodują, że nie widać już tak wyraźnie poszczególnych punktów ekranu. Niestety, nie zmienia to jednak faktu, że ręczne rysowanie oprócz zalet ma wady. Największy mankament sprawia prosta zależność, że im więcej kolorów i większa rozdzielczość tym dłużej trwa rysowanie. Związane są z tym wydatki albo na dodatkowy sprzęt (bardziej ergonomiczne niż myszka urządzenia do rysowania) lub na lekarza (jeszcze kilka lat temu graficy z firmy Sierra On-Line, pracujący wiele godzin na dobę często skarżyli się na bóle i zapalenia stawów rąk). Inna wada ręcznego rysowania to potrzeba częstego retuszowania czyli dokonywania poprawek, najlepiej w powiększeniu.

### Computer Aided Design

Skrót CAD oznacza po polsku projektowanie wspomagane komputerowo. Chodzi po prostu o tworzenie na ekranie różnych geometrycznych tworów, które czasami przydają się inżynierom-projektantom. Metodę tą wykorzystuje się też do prezentacji struktur przedmiotów, a często w poważnych symulatorach do szkolenia (pilotów, kierowców, itd.) i ich rozrywkowych odpowiednikach czyli grach. Nie jest to wcale najgorszy sposób tworzenia grafiki, bo przecież z brył i figur płaskich można złożyć wiele różnych rzeczy. Przykład: rysunek ma przedstawiać jakiś konkretny samochód widziany pod z przodu pod pewnym kątem. Można go narysować ręcznie, zeskanować ze zdjęcia, zdigitalizować z kamery, ale można też skonstruować z kilku brył w odpowiednim programie i obracając w dowolnym kierunku ustawić nie tylko pod tym kątem, który sobie założyliśmy, ale pod trochę innym, z którego samochód będzie lepiej wyglądał. Wadą przy tej metodzie jest duża liczba obliczeń (przy skomplikowanych obiektach) i pewne ograniczenia w wyglądzie (wszystkie kształty o regularnych krawędziach, kanciastość obłych przedmiotów) oraz potrzeba przestrzennego, matematycznego myślenia. Pocieszeniem są niewielkie w sumie wymagania sprzętowe (poza dość szybkim komputerem) oraz łatwy dobór i zmiana parametrów.

### Raytracing

To słowo jest połączeniem angielskiego ray czyli promień z równie angielskim tracing czyli śledzenie. Oznacza to metodę tworzenia grafiki przez śledzenie promieni czyli badanie przez program jak ma wyglądać każdy wcześniej zdefiniowany obiekt, na który pada światło, jaki stopień tego światła odbija, jaki rzuca cień, jaki posiada kolor, itd. Najczęściej do raytracingu korzysta się z gotowych brył, które ustawia się w określonym miejscu przestrzeni (w zdefiniowanym układzie współrzędnych) i po określeniu parametrów takich jak jasność źródła światła czy stopień chropowatości obiektów następują obliczenia końcowego efektu.

Wady raytracingu to potrzeba długich obliczeń, przy dużej rozdzielczości i liczbie kolorów (co jest warunkiem niezłych efektów) mogą one trwać nawet wiele dni (choć to nieprawdopodobnie brzmi, także na TT, która ma procesor 68030 i koprocesor!), z reguły jednak nie przekraczają kilku czy kilkunastu godzin. Wadą, dla niektórych wszakże zaletą, jest idealność rysunku wytworzonego tą techniką. Brak nieregularnych powierzchni i matematyczna doskonałość każdej krawędzi może się oczywiście podobać i mieć wiele zastosowań, ale

wciąż w programach do raytracingu dąży się do uzyskania efektów, które zbliżałyby czysto komputerowe obrazy do tych z rzeczywistości.

### Fraktale i wykresy

Już powyższa metoda wykazała, że matematyka jest bardzo przydatna do tworzenia grafiki. Tam jednak służyła do obliczeń wzajemnych relacji między obiektami, a przecież można same obliczenia prezentować w formie graficznej. Od pomysłu do realizacji nie minęło wiele czasu i dzisiaj można oglądać mnóstwo programów do rysowania wykresów, od prostych funkcji, aż po skomplikowane wzory, których wynikiem są bajeczne efekty graficzne. Najpopularniejsze są fraktale, dla których wciąż poszukuje się nowych zastosowań (ostatnio wymyślono rewelacyjną metodę kompresji grafiki za pomocą fraktali). Są to twory matematyczne, które powstają według pewnego schematu, wzoru. Podobnie jest też ze zwykłymi wykresami funkcji, które może nie są tak efektowne jak fraktale, ale mogą być miłe dla oka. Zastosowanie fraktali i wykresów do tworzenia grafiki jest trochę mniej popularne niż innymi sposobami, bo mimo wszystko temat obrazu musi być ograniczony, ale ma swoje zalety. Nie potrzebny jest wyrafinowany sprzęt, łatwo zmienia się wszelkie parametry i nie trzeba mieć wielkich zdolności, by stworzyć jakiś ciekawy wzór do narysowania.

### Inne

Inne nie znaczy gorsze. Chodzi bowiem o nietypowe tworzenie grafiki. Do takich niekonwencjonalnych metod zaliczam „przeliczanie" muzyki na obraz. Brzmi to absurdalnie, ale naprawdę jest możliwe. Polega to na tym, że wszystkie elementy muzyki takie jak nuty czy pauzy mają swój graficzny odpowiednik (np. cała nuta to kolor czerwony, ćwierć nuta z kropką to zielony kwadrat) i podczas gry na ekranie komputera tworzy się obraz muzyki. Wtedy można np. zaobserwować, czy bardziej niebieskie jest Bolero Ravela czy najnowszy przebój Madonny.

Mniej zaskakującym sposobem jest wykorzystywanie programów astronomicznych. Wystarczy w takim programie wybrać wycinek nieba o dowolnie wybranej porze i zaprezentować na ekranie. Jeśli np. gwiazdy są ciekawie ułożone, albo Księżyc wygląda ładnie to grafika jest już gotowa.

Do tworzenia dobrej grafiki można też wykorzystać tekst. Znów brzmi to dziwnie, ale nie chodzi tym razem o „tłumaczenie" tekstu na obraz. Mam tu na myśli obrazy złożone ze znaków. Jeśli ktoś dalej nie wie o co chodzi, niech kupi sobie drukarkę, która może drukować tylko znaki i wymyśli jak można wydrukować na niej grafikę. Oczywiście tylko w jeden sposób, w którym obraz nie będzie składał się z punktów, ale z liter. Na przykład prostokąt będzie wyglądał tak:

```
zzzzzzzzzzzzzzz
z             z
z             z
zzzzzzzzzzzzzzz
```

To samo można zrobić na ekranie komputera, choć obrazki będą ograniczone przez maksymalną ilość znaków w wierszu i kolumnie.

### Łączenie

Oczywiście nie jest powiedziane, że nie da się łączyć metod rysowania. Na przykład można zeskanować kształt samochodu za pomocą skanera do brył, przerobić go nieco w programie typu CAD, potem zrobić raytracing, a na końcu poprawić kilka rzeczy ręcznie. Trzeba stwierdzić, że takie mieszanie technik uzyskiwania obrazu jest nawet lepsze i daje często bardziej pożądane efekty.

Na koniec chciałbym poinformować, że zmieniono tytuł mojego artykułu z nr 3/92 z „Formaty graficzne" na „TIF, GIF... PIF, PAF" bez mojej zgody i wiedzy.

*[W zasadzie moglibyśmy napisać, że redakcja zastrzega sobie prawo do..., ale nie o to chodzi. Więcej się to może nie powtórzy [he, he...]*

## nadal trwa...

## Konkurs

**Jeżeli komuś nie wystarcza własna satysfakcja z wykonanej grafiki lub animacji, niech pochwali się na szerszym forum. Wszystkie dzieła mile widziane, można je przysyłać na mój (niżej podany) adres. Jeżeli nadejdzie ich więcej, będzie można zorganizować konkurs. Nagrody, poza publikacją nazwiska zwycięzcy w ST^E^fanie, nie są wykluczone. Oczywiście prawa autorskie będą w pełni respektowane, zastrzegam sobie jedynie prawo do publikacji nadesłanych prac w ST^E^fanie i na dyskietkach dodawanych do ST^E^fana. I jeszcze jedno. Animacje muszą być zapisane albo jako programy samouruchamialne (PRG, TOS, itd.), albo w formacie IFF (Deluxe Paint), SEQ i DLT (Cyber Paint) lub SCR (Aegis Animator). Z innymi formatami mogą być kłopoty.**



**ul. Wojrowicka 46/26, 54-436 Wrocław.**

Konkurs

# Errata

Przygotowując dyskietkę do ostatniego numeru ST^E^fana popełniliśmy pewien, fatalny w skutkach, błąd. Mianowicie nie sprawdziliśmy czy nowa wersja (2.1) programu Inkaust (a zatem również program demonstracyjny dołączony do ST^E^fana) pracuje na komputerze z pamięcią poniżej 2 MB. Chociaż wiedzieliśmy, że w tej wersji poprawiony (i powiększony) został słownik, jak również byliśmy pewni, że stara pracowała na 1040 ST, jakoś uszło naszej uwadze sprawdzenie Inkaustu 2.1 Demo właśnie pod tym kątem. W wyniku tego każdy, kto uruchamiał Inkaust Demo na komputerze poniżej 2 MB RAM otrzymywał w czasie ładowania komunikat o błędzie, po czym komputer powracał do systemu. Powyższa sytuacja wynikała z informacji zawartej w pliku konfiguracyjnym programu, z której wynikało, że ma on w trakcie startu ładować również słownik, którego wielkość była powodem problemów.

Obecnie na dyskietce zamieszczamy taki plik konfiguracyjny, że umożliwia (po zastąpieniu nim starego) na uruchomienie Inkaustu (bez załadowanego słownika).

Mamy nadzieję, że teraz wszystko okaże się w porządku.

*Redakcja*

*(nad całością męczył się, jak zwykle, spk. (NRG)) — as. Przemek Kobel*

# FX 4,5 — The Smell of Silicon

Na dobry początek chciałbym wyjaśnić „nadmiar" listingów w asemblerze w poprzednim wydaniu FX-ów. Najwyraźniej wystąpiło „ssanie" dysku twardego (pewnie ktoś potrzebował miejsca...). Efekt tego jest taki, że program w GFA Basicu wykorzystujący procedurę nr 1 zmienił fazę w sposób cichy, choć nagły. Ze swej strony proponuję „własnoręczne" wykorzystanie procedury (w sposób opisany w poprzednim artykule i pozostałych listingach). Na szczęście, najciekawsze programy pozostały nietknięte.

A teraz, jak ostatnio obiecałem, dowalimy Shifterowi tak, że aż zaświszczy (niestety, procesorowi też się dostanie). Na wstępie nieco teorii.

Jak wiadomo, obraz generowany przez Atari ST dla monitora RGB, lub telewizora składa się z 200 linii wyświetlanych 50 razy na sekundę. Cała powierzchnia ekranu to nieco ponad 312 linii (za Wielką Wodą — 256 linii, 60 obrazów/sek.). Linie te wyświetlane są od góry do dołu z szybkością ponad 15 tys. razy/sek. Plamka tworząca takie linie wędruje od lewej strony do prawej zmieniając co jakiś czas kolor. Zmienianiem koloru plamki w Atari ST zajmuje się nasza ofiara — Shifter. W niskiej rozdzielczości ST wykonuje on następujące operacje:

— pobranie z pamięci obrazu czterech słów (8 bajtów)

— 16 operacji przesunięcia każdego z tych słów; wypadające bity tworzą adres rejestru, z którego pobierane są wartości przesyłane do przetworników generujących sygnały koloru.

Cykl ten powtarzany jest aż do momentu, gdy plamka osiągnie obszar ramki. Dla tego obszaru dane do wyświetlana pobierane są z rejestru 0.

Programista ma dostęp tylko do pamięci, z której pobierane są dane, oraz do rejestrów kolorów układu Shifter — nie można zmienić danych znajdujących się już w „obróbce".

Naszym celem jest zwiększenie ilości kolorów. Manipulacja na pamięci obrazu nic nie da — zmieni się tylko kolejność wyświetlanych kolorów, a nie ich ilość. Jedynym wyjściem jest odpowiednio szybkie zmienianie zawartości rejestrów kolorów układu Shifter. Najlepszym językiem programowania do takich zastosowań jest język maszynowy (największa prędkość).

Użycie procesora MC 68000, 8 MHz umożliwia zwiększenie liczby kolorów do 48 w jednej linii, każda linia może mieć inny zestaw kolorów. Idea programu realizującego taki efekt jest następująca: plamka znajduje się na początku linii; w rejestrach kolorów znajduje się 16 różnych definicji kolorów, więc pierwszych 16 punktów można uznać za całkowicie niezależne; w tym momencie program zaczyna zmieniać po kolei wszystkie rejestry kolorów; w czasie zmiany zawartości każdego rejestru plamka „przelatuje" stały odcinek mierzony np. w pixlach, a więc co kilka punktów obrazu w dostępnej palecie 16 kolorów zmieniony zostaje jeden kolor. Okazuje się, że czas potrzebny na trzykrotną zmianę całej palety dokładnie pokrywa się z czasem wyświetlania jednej linii. Jedyne, co należy zrobić, to odpowiednio zsynchronizować moment startu procedury przepisującej kolory, (no i przygotować pamięć obrazu, oraz dane dla procedury). Efektem ubocznym takich zabaw jest gwałtowny spadek wydajności procesora (teoretycznie — ok. 60 % strat; obecnie stosowane procedury pożerają 75 % czasu procesora z powodu pętli oczekującej, która wykonywana jest na początku każdego przerwania pionowego). Jeśli przyjąć, że czas potrzebny na utworzenie całego obrazu na ekranie monitora = 100 % czasu procesora, to procedury będące obecnie w użyciu pochłaniają obszar od początku obrazu (początek górnej ramki), aż do końca ostatniej linii. Gdy plamka na ekranie kineskopu skończy rysować ostatnią linię obrazu zawierającą jakąś treść, nasza procedura kończy swą pracę i oddaje czas reszcie systemu.

Rozwiązanie to jest wykorzystywane przez program Spectrum Paint. Mieszanka tego efektu i pierwszego efektu z poprzedniego wydania FX-ów zastosowana jest w programie Quantum Paint, natomiast jednoczesne wykorzystywanie procedury podobnej do dziś prezentowanej oraz uproszczonej wersji programu nr 2 (z FX nr 4) daje nam możliwości takie, jak w programie Photochrome (96 kolorów w linii z palety „4096" kolorów — w rzeczywistości paleta obejmuje 3375 kolorów). Jak widać, parametry obrazu naszego STFM mogą wyglądać całkiem nieźle (zwłaszcza, gdy się weźmie pod uwagę fakt, że mamy do czynienia z ponad 8-letnim komputerem). Ze swojej strony mogę dodać, że liczba 96 kolorów w linii nie jest jeszcze liczbą maksymalną...

Na koniec mały opis formatu obrazków *.SPU (Spectrum Paint): 0–31999 (dec) — mapa bitowa obrazu (32000 bajtów) 32000–51103 (dec) — paleta kolorów obrazu (19104 bajtów — 199 linii x 3 palety w linii x 32 bajty na jedną paletę)

P.S. Zapomniałem o ST^E. Dla tego komputera maksymalne osiągi na dziś, to 96 kolorów w linii z palety „32768" (29791) kolorów.

A teraz, jak mawiają kolesie z mojego ulubionego serialu-tasiemca, DIE WELL!

```
* FX 4.5 - listing w assemblerze
* Spectrum Display (50 Hz)
* (opr. na podst. proc. Muttera Frédérica przez spk (NRG))
start
 lea.l spu_display(pc),a0 ; adres procedury przerwania do
                          ; A0
 lea.l oldvec(pc),a2 ; miejsce na stary wektor
 movea.l vblvec,a1 ; wektor przerwania VBL do A1
 cmpa.l a0,a1 ; porównanie wektorów
 beq deinstal ; jeśli równe to powrót do normy
instal
 move.l a1,(a2)+ ; zapamiętanie starego wektora
 move.l 8(sp),(a2)+ ; zapamiętanie adresu obrazka
 move.l a0,vblvec ; ustawienie nowego wektora VBL
 rts
deinstal
 move.l (a2),vblvec ; przywrócenie starego wektora VBL
 rts
spu_display ; procedura obsługi przerwania VBL
 move.w sr,-(sp) ; zapamiętanie starego rejestru SR
 ori.w #$0700,sr ; zablokowanie przerwań
 lea.l $FFFF8240,a4 ; adres palety kolorów do A4
 lea.l $FFFF8209,a5 ; adres LSB licznika pamięci obrazu
 lea.l start_sync(pc),a6 ; adres opóźnienia do A6
 move.l picadr(pc),a3 ; adres obrazka do A3
 move.l a3,d0
 ror.l #8,d0 ; przepisanie go do rejestrów
 move.b d0,$FFFF8203
 ror.l #8,d0
 move.b d0,$FFFF8201
 adda.l #32000,a3 ; ustalenie adresu palety kolorów
 move.w #198,d7 ; procedura obsługuje 199 linii
 moveq.l #0,d0 ; zerowanie rejestru D0
wait
 tst.b (a5) ; sprawdzenie, czy SHIFTER zaczął
 beq wait ; pobierać dane
 move.b (a5),d0 ; jeśli tak, synchronizowanie pętli
 adda.l d0,a6 ; zmiana momentu wejścia
 jmp (a6) ; skok w pętle wyświetlającą
start_sync
 REPT 81 ; 81 rozkazów NOP (opóźnienie)
 nop
 ENDR
nextline ; początek linii
 nop ; nop (dla 50 Hz)
 lea.l (a4),a0 ; adres rejestrów palety do A0-A2
 lea.l (a4),a1
 lea.l (a4),a2
 move.l (a3)+,(a0)+ ; Paleta nr 1
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a0)+
 move.l (a3)+,(a1)+ ; Paleta nr 2
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a1)+
 move.l (a3)+,(a2)+ ; Paleta nr 3
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 move.l (a3)+,(a2)+
 nop
 dbra d7,nextline
 move (sp)+,sr ; odtworzenie starego SR
skok
 jmp $00000000
oldvec=skok+2
picadr
 ds.l 1
vblvec=$4ce
```

```
.Tu wczytać plik "fx45.inl" (HELP, L, plik po assemblacji
(fx45.inl))
INLINE display%,1024
RESERVE 20480 ! zwolnienie pamięci
oldlog%=XBIOS(3) ! zapamiętanie starych
oldfiz%=XBIOS(2) ! parametrów obrazu
oldrez%=XBIOS(4)
pic%=MALLOC(51104) ! pamięć na obrazek
XBIOS(5,L:pic%,L:pic%,W:0)
OPEN "I",#1,"a:\fx45\pic.spu" ! odczyt obrazka
BGET #1,pic%,51104
CLOSE #1
XBIOS(38,L:display%+&H1C,L:pic%) ! włączenie procedury
wyświetlającej
INP(2) ! program użytkownika
XBIOS(38,L:display%+&H1C,L:pic%) ! wyłączenie procedury
wyświetlającej
XBIOS(5,L:oldlog%,L:oldfiz%,W:oldrez%) ! odtworzenie
starych par. obrazu
MFREE(pic%) ! zwolnienie pamięci
END ! koniec
```

Dariusz Pilarczyk

# Automaty komórkowe

**Nie wiem, czy wszyscy czytelnicy ST[E]fana mają podobne odczucia jak ja, ale mi osobiście dokucza mała ilość programów ukazujących się w naszym ulubionym piśmie o Atari.**

Osobnik taki jak ja, który z programowaniem ma już do czynienia parę lat i, prawdę mówiąc, jest już nim lekko znudzony, szuka, jak mi się wydaje, w czasopismach przede wszystkim tekstów niedużych, ale efektownych programów.

A co programista znajdzie dla siebie w ST[E]fanie? Jest F/X — przeważnie na niską rozdzielczość (nie każdy model ST ma modulator TV — mój na przykład nie!), jeden lub dwa inne programiki, no i „seriale" o GEMie i TOSie. Jak na dwa miesiące, jakie upływają między kolejnymi numerami, nie jest to dużo.

Dla wszystkich, którzy odczuwają niedosyt, spróbowałem przygotować jakiś program i napisać o czymś, co być może ich zainteresuje i być może zachęci do samodzielnych eksperymentów. Czy mi się udało — oceńcie sami.

Temat jest modny m. in. w fizyce, ale także w grafice komputerowej. Ja znam go (trochę) z wykładów prowadzonych przez dr M. Wolfa z IFT UWr, dlatego artykuł będzie trochę nawiązywał do fizyki (i chemii).

Najpopularniejszym przykładem automatu komórkowego jest *Gra w życie*. Myślę, że większość Czytelników zetknęła się już kiedyś z tą zabawą. Na wszelki wypadek przedstawię (przypomnę?) jej zasady:

Wyobraźmy sobie prostokątną płaszczyznę podzieloną na kwadratowe pola — komórki. Każda taka komórka sąsiaduje z ośmioma innymi. W danej chwili w jednej komórce może przebywać co najwyżej jeden robaczek: rodzi się on w niej, mieszka tam, i po pewnym czasie na ogół umiera. Po jego śmierci komórka staje się wolna — może w niej zamieszkać ktoś inny. O narodzinach i przetrwaniu decydują trzy następujące reguły:

1. Jeśli pewna komórka jest pusta, lecz sąsiaduje z trzema komórkami, w których przebywają jakieś robaczki, to rodzi się w niej nowy robaczek.

2. Jeżeli komórka, w której mieszka pewien robaczek, sąsiaduje z co najmniej czterema innymi zamieszkanymi, to robaczek umiera wskutek przeludnienia.

3. Jeśli nasz robaczek ma mniej niż dwóch sąsiadów to umiera z osamotnienia.

Zasady te są bardzo proste, ale prowadzą do ciekawych wyników. Dobierając różne rozkłady początkowe zamieszkanych komórek otrzymuje się rozmaite twory, które poruszają się po płaszczyźnie, pożerają sąsiadów, rozrastają się do olbrzymich rozmiarów bądź po pewnym czasie zanikają. Niektóre konfiguracje prowadzą do stanów stabilnych, które nie ulegają dalszym przeobrażeniom. Są też takie, które po przejściu pewnej liczby pośrednich stadiów przybierają na powrót swą wcześniejszą postać. Zresztą, zainteresowanym radzę, aby sprawdzili sami. Napisanie odpowiedniego programu nie jest trudne, można też poszukać czegoś odpowiedniego w oprogramowaniu public domain. My zajmiemy się trochę innymi automatami. Piszę o nich w liczbie mnogiej, bo będzie ich 256.

Najpierw teoria i mnóstwo szczegółów technicznych. Automaty komórkowe są w pewnym sensie matematyczną idealizacją układów np. fizycznych, biologicznych, w których czas i przestrzeń oraz inne obserwowane wielkości przyjmują wartości dyskretne. „Dyskretną przestrzeń" możemy sobie wyobrażać jako „zwykłą przestrzeń podzieloną na komórki" — w *Grze w życie* są to komórki kwadratowe, ale często stosuje się także komórki sześciokątne czy trójkątne. Całkowita liczba komórek N jest skończona i stała (nie zmienia się podczas ewolucji automatu). Każda taka komórka może się znaleźć w jednym ze skończonej liczby stanów, które będziemy przeważnie oznaczac liczbami naturalnymi: 0, 1, 2 itd. W najprostszym przypadku mamy dwa stany: 0 i 1. Tak jest np. w *Grze w życie*, gdzie 0 utożsamilibyśmy ze stanem „pusta komórka" a 1 ze stanem „komórka zamieszkana".

Stan początkowy automatu, czyli stany wszystkich komórek w chwili początkowej $t=0$, zadaje się z góry np. stosując rozkład losowy albo według widzimisię kogoś. Stan w chwili $t+1$ oblicza się w oparciu o stany komórek w chwili poprzedniej $t$ stosując pewne ustalone wcześniej reguły (reguły ewolucji automatu).

W pewnym sensie każdy model komputerowy jakiegoś zjawiska, czy nawet sam komputer — to automaty komórkowe.

Jeżeli zabawa odbywa się na płaszczyźnie, tak jak w *Grze w życie*, to mówimy, że mamy do czynienia z dwuwymiarowym automatem komórkowym. Oczywiście, można konstruować automaty jednowymiarowe, trójwymiarowe itd. Te jednowymiarowe, którymi się teraz zajmiemy, są zbudowane z komórek ułożonych wzdłuż prostej, sąsiadujących ze sobą parami. Nie jest to niezbędne, ale dla wygody często się przyjmuje, że komórka 1 sąsiaduje z komórką N-tą. Dzięki temu będziemy mogli traktować komórki skrajne tak samo, jak pozostałe.

Oznaczmy przez $a_i^{(t)}$ stan $i$-tej komórki w chwili $t$. Może on przyjmować wartości z przedziału $0, 1, ..., k-1$, gdzie $k$ jest liczbą dopuszczalnych stanów komórki. Definicję naszego 1-wymiarowego automatu zapiszemy tak:

$$a_i^{(t+1)} = f(a_{i-l}^{(t)}, a_{i-l+1}^{(t)}, ..., a_i^{(t)}, ... a_{i+r-1}^{(t)}, a_{i+r}^{(t)})$$

Wzór ten mówi, że stan $i$-tej komórki w chwili $t+1$ można obliczyć za pomocą pewnej funkcji $f$, znając stany komórek o numerach $i-l, i-l+1, ..., i, ..., i+r-1, i+r$ w chwili $t$. Liczby $r$ i $s$ są tutaj pewnymi stałymi, a funkcja $f$ pełni taką samą rolę jak reguły narodzin i śmierci robaczków w *Grze w życie* — będziemy ją więc nazywać regułą. Dobierając różne $f$, $r$ i $s$ możemy konstruować różne automaty.

Szczególnie proste są tzw. automaty elementarne, dla których $k=2$ (tj. dopuszczamy tylko dwa stany komórek: 0 lub 1) a ponadto stan komórki w następnej chwili zależy tylko od jej stanu i stanu obu jej najbliższych sąsiadów w chwili poprzedniej (czyli $r=s=1$). Mamy osiem możliwych kombinacji stanów tych komórek:

**111 110 101 100 011 010 001 000**

Aby zdefiniować regułę, powinniśmy podać dla każdej z tych kombinacji liczbę 0 lub 1. Oznacza to, że potrafimy ją zakodować na jednym bajcie. (Innymi słowy istnieje 256 reguł dla automatów elementarnych).

Przykładowo reguła

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

odpowiada liczbie dwójkowej $(00001101)_2$ czyli 13 dziesiętnie.

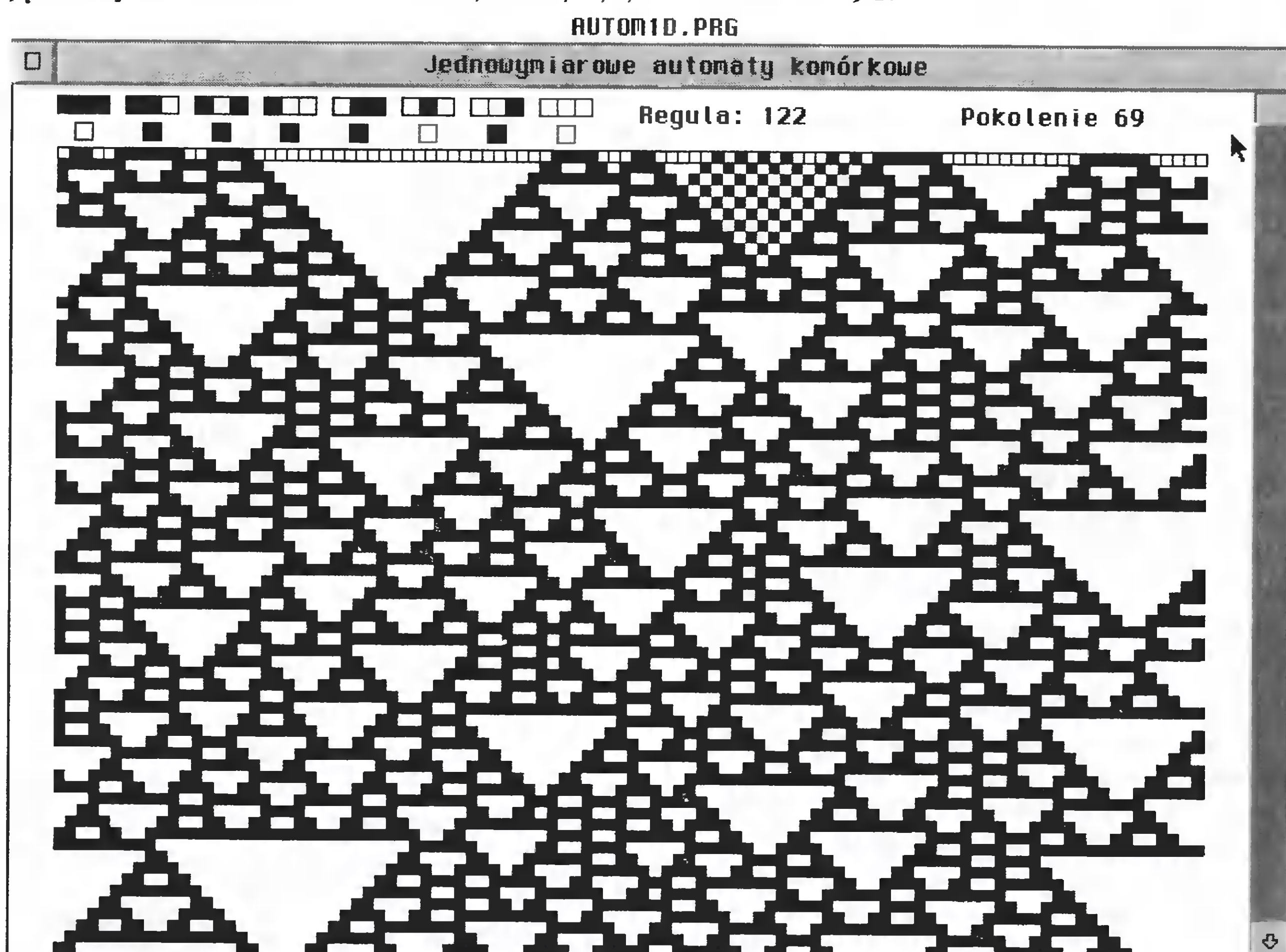
Podobnie reguła określona wzorem

$$a_i^{(t+1)} = (a_{i-1}^{(t)} + a_{i+1}^{(t)}) \text{ MOD } 2$$

ma numer 90, bo ma taką „reprezentację bitową":

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

a prosty rachunek daje nam $(01011010)_2 = 64+$ $+16+8+2 = 90$.



AUTOM1D.PRG

Reguła tożsamościowa (taka, która „nic nie zmienia")

```
111 110 101 100 011 010 001 000
 1   1   0   0   1   1   0   0
```

to reguła 204.

Na listingach obok znajdują się dwa moduły w języku C tworzące program do badania własności tych automatów. Program główny umieszczony jest na listingu 1. Moduł z listingu 2 o nazwie „KOMÓRKA" zawiera kilka funkcji pomocniczych. Można go wykorzystywać także przy tworzeniu innych automatów. (Jeśli ten artykuł doczeka się jakiegoś „dalszego ciągu", to tak się stanie.)

Nasz program powinien działać we wszystkich trybach graficznych. W niektórych trybach graficznych Atari TT i Falcona obraz może mieć nieprawidłowe proporcje — komórki na ekranie nie będą kwadratowe. W takich wypadkach radzę posłużyć się funkcją XBIOS EgetShift. Mój kompilator (Turbo C 2.0) niestety nie zezwala na jej użycie (a w każdym razie nigdzie nie udało mi się znaleźć jej prototypu). O proporcjach boków komórek decyduje instrukcja

```
dy = (getrez()==SREDNIA__ROZDZ)?
dx/2 : dx;
```

zmienną dy należy dobrać tak, aby prostokąt o bokach dx w kieruku osi X i dy w kierunku osi Y wyglądał na ekranie jak kwadrat.

Po uruchomieniu programu na ekranie otrzymujemy obrazek przedstawiający ewolucję automatu. Kolejne wiersze to kolejne pokolenia komórek. W lewym górnym rogu znajduje się reguła przedstawiona w postaci „bitowej" oraz w postaci kodującej ją liczby. Po prawej stronie pokazywany jest numer najmłodszego pokolenia widocznego na ekranie. Regułę możemy zmieniać za pomocą myszki (zapalając lub gasząc odpowiednie bity), badź też wklepując nową regułe z klawiatury. Klawisz R nadaje komórkom zerowego pokolenia przypadkowe wartości, natomiast J ustawia środkową na 1, a pozostałe na 0. Stany poszczególnych komórek można modyfikować także przy pomocy myszy. Jeśli zechcemy przewinąć ekran w dół używamy pola przesuwu okna o stronę w dół lub naciskamy klawisz spacji (obie te operacje przewijają obraz o „ekran minus dwa wiersze"). Do dużych „skoków w czasie" przydatny może być suwak, który daje nam przesunięcie z zakresu 0..1000 pokoleń natomiast strzałka suwaka (jedno pokolenie) pełni w praktyce jedynie funkcję dekoracyjną. Wyjście z programu odbywa się za pomocą pola zamykającego okno (zamykacza). Klawisz S na przemian włącza i chowa pomocniczą siatkę, która może być przydatna przy analizie ewolucji układu.

Aha, program jest niezbyt szybki. Winą za to należy obarczać nie algorytm (chociaż można by go przyspieszyć, np. kopiowania tablic w funkcji ewol1() można było uniknąć), ale funkcje graficzne VDI. Jeśli ktoś wie, jak można poprawić szybkość, to może zechciałby się podzielić z innymi?

Jeżeli do tej nie uruchomiliście programu, zróbcie to teraz. Myślę, że rozumiecie, dlaczego wspominałem o grafice komputerowej? Radzę poeksperymentować z innymi regułami, np. 11, 57, 73, 99, 129, 151.

Szereg ciekawych (jeśli kto woli — ładnych, zwłaszcza w wysokiej rozdzielczości i dla dużego N — powiedzmy 610) obrazków wygenerujemy, jeśli w pokoleniu początkowym pozostawimy tylko jedną komórke zaczernioną. Wśród nich bardzo popularny jest kształt generowany przez wspomnianą już regułę 90. Naciskamy klawisz J, a następnie piszemy 90 i naciskamy Enter. Rysuje się... dywan Sierpińskiego.

Z moich eksperymentów wynika, że fraktal ten ma w świecie automatów komórkowych nader liczną rodzinkę.

Oprócz, tego, że obrazek ten możemy otrzymać także z pomocą innych reguł, co nie jest zbyt ciekawe, to nietrudno znaleźć szereg reguł generujących fraktale trochę inne, lecz wyraźnie podobne. Zresztą, najpierw pomyślmy.

Reguła 90, jak pamiętamy określa nam stan w i-tej komórce w nowym pokoleniu jako sumę modulo 2 stanów jej lewego i prawego sąsiada w poprzednim pokoleniu. Co się stanie, jeśli zamiast lewego sąsiada damy stan i-tej komórki (reguła 102)? Sprawdźmy. Otrzymujemy też dywan Sierpińskiego, ale inny. Jak wygenerować obrazek symetryczny domyślacie się zapewne sami. Teraz proponuję wprowadzić regułe 126. Znowu coś dywanowatego. Jest ona jakby „matką" omówionych poprzednio reguł. Jeśli wyzerujemy w niej bity 3 i 4, otrzymamy zwykła regułe 90, a jeśli wyzerujemy bity 1 i 6 lub 2 i 5 — dywany „połówkowe". Możemy też zanegować wszystkie jej bity: daje to regułę 129. „Oczywiście" powinniśmy otrzymać negatyw obrazka (inwers). Czy 129 ma własności analogiczne do 126? Sprawdzcie. Jeśli chcecie mieć fraktala w innym kolorze niż tło, użyjcie reguły 182.

Są jeszcze „dalecy krewni": 150, 169. Ten ostatni zachowuje się niemal jak istota żywa: porusza się w lewo i bardzo powoli rośnie. Po około 2048 pokoleniach jest „już" tak duży, że obejmuje wszystkie komórki automatu. Nie wychodzi mu to na zdrowie: gwałtownie obumiera, pozostawiając po sobie kilka komórek, z których powstaje jego następca i cykl się powtarza).

Radzę poeksperymentować samodzielnie. Automaty komórkowe są dziedziną słabo zbadaną, która nie doczekała się jeszcze dobrej teorii. Wiele zależy od Waszej pomysłowości.

Teraz jeszcze trochę o własnościach automatów komórkowych. Łatwo zauważyć, że ewolucja dowolnego automatu jest okresowa, z okresem co najwyżej $2^N$. Na ogół jest to jednak tak duża liczba, że okresowość ta często nie jest widoczna. Ewolucja niektórych automatów komórkowych (niekoniecznie jednowymiarowych) wydaje się wręcz nie wykazywać żadnych prawidłowości — czyżby chaos, słowo modne w fizyce?

Inną własnością jest nieodwracalność: jeśli się ma jakiś stan, to na ogół nie da się ustalić, z jakiego stanu on powstał. Fizycy mówią także o braku ergodyczności — oznacza to, że nie wszystkie stany są odwiedzane w trakcie ewolucji automatu.

Automaty komórkowe często wykazują małą podatność na zaburzenia warunków początkowych: zmieniając nieco stan początkowy, np. w tylko jednej komórce przeważnie nie wywołamy jakichś gwałtownych zmian ewolucji. Dla niektórych automatów wpływ zaburzeń ma tendencje do zanikania. Własność ta przypomina organizmy żywe. Jest wykorzystywana także m. in. w sieciach neuronowych, o których może kiedyś opowiem.

Na zakończenie dodam przykłady dwóch automatów dwuwymiarowych, używanych do całkiem poważnych badań.

Pierwszy dotyczy reakcji chemicznych z udziałem katalizatora. Jak zapewne pamiętacie, na chemii pisało się równania reakcji tego typu:



$$A + B \xrightarrow{kat} AB$$

Równanie wygląda prosto (reakcja zachodzi, i już), w praktyce bywa różnie. Jeśli np. stężenie jednego ze składników będzie za duże, może on pokryć powierzchnię katalizatora (chemicy mówią o „zatruciu katalizatora"). Nie wierzycie? Sprawdzcie. Budujemy w tym celu automat komórkowy oparty na sieci kwadratowej. Sieć ta (jej komórki) pełnią rolę powierzchni katalizatora. Dopuszczamy cztery stany komórek:

- 0 — komórka pusta
- A — komórka zawiera atom substancji A
- B — komórka zawiera atom substancji B
- C — w następnej chwili komórka być może znajdzie się w stanie B, jeśli będa spełnione pewne warunki (patrz reguły)

Reguły ewolucji natomiast są następujące (dlaczego takie — pytajcie chemików):

1. Jeżeli $a_i^{(t)} = 0$, to $a_i^{(t+1)} = A$ z prawdopodobieństwem $X_A$ ($X_A$ — koncentracja atomów substancji A) lub $a_i^{(t+1)} = C$ z prawdopodobieństwem $X_B = 1 - X_A$.

2. Jeżeli $a_i^{(t)} = A$ to jeżeli w co najmniej jednym najbliższym sąsiedzie jest atom B, to $a_i^{(t+1)} = 0$, w przeciwnym razie $a_i^{(t+1)} = A$.

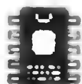
3. Jeżeli $a_i^{(t)} = B$ to jeśli co najmniej jeden z najbliższych sąsiadów jest w stanie A, to $a_i^{(t+1)} = 0$, w przeciwnym razie $a_i^{(t+1)} = B$.

4. Jeżeli $a_i^{(t)} = C$ to jeśli żaden z najbliższych sąsiadów nie był też w stanie C, to $a_i^{(t+1)} = 0$, w przeciwnym razie $a_i^{(t+1)} = B$. $X_A$ jest tutaj stałą (i oczywiście $X_B$ też). Zmieniając ją, możecie uzyskać efekty, o których pisałem. Zainteresowani mogą sporządzić wykres zależności stężeń atomów A i B na powierzchni katalizatora. Jest to pracochłonne, ale chemicy potrafią z niego sporo odczytać.

Drugi automat jest chyba prostszy. Jego angielską nazwę można przetłumaczyć jako „piramida piasku". Własności takiej piramidy wszyscy znają: jeśli nasypiemy za dużo piasku (czyli jeśli wysokość piramidy w pewnym miejscu przekracza pewną wartość graniczną H), to się obsypuje. Niech z(x, y) oznacza wysokość słupa piasku w punkcie o współrzędnych (x, y), L — długość boku podstawy piramidy. Na brzegach piramidy będziemy utrzymywać wysokość równą zero. Oto idea automatu komórkowego:

```
for x=1 to L do
{
   for y=1 to L do
      jeśli(z(x, y)>=H)
      {
         z(x, y) = z(x, y) - 4;
         z(x+1, y) = z(x+1, y) + 1;
         z(x-1, y) = z(x-1, y) + 1;
         z(x, y+1) = z(x, y+1) + 1;
         z(x, y-1) = z(x, y-1) + 1;
      }
   z(0, x) = z(x, 0) = z(x, L) = z(x,
   L) = 0;
}
jesli(dla każdego x,y mamy z(x,y)<H)
   stan jest statyczny, STOP;
```

To, co się otrzyma, nazywa się samozorganizowanym stanem krytycznym. Jest on niestabilny — jeśli dorzucimy jedno ziarno w losowym punkcie z(x, y) = H+1, to nastąpi przesypywanie się piasku. Czas potrzebny, aby układ się zrelaksował (powrócił do stanu statycznego) ma bardzo duży rozrzut od 1 do tysięcy. ❑



Witold Warczak

# TOS od środka część 4

**Tematem ostatniego artykułu z cyklu „Tos od środka" będą funkcje GEMDOS-u. GEMDOS zawiera niezależne od sprzętu funkcje wejścia-wyjścia, odpowiada za zarządzanie pamięcią i obsługę plików.**

Przy korzystaniu z funkcji GEMDOS-u z poziomu asemblera należy pamiętać, że wywołanie ich zmienia zawartość rejestrów D0–D1 i A0–A1 procesora, zaś wyniki są zawsze zwracane w rejestrze D0. Użytkownicy GFA Basica mogą odetchnąć z ulgą — większość funkcji GEMDOS-u jest dostępna jako rozkazy tego języka.

**0 Pterm0**

C:

```
void Pterm0( void );
```

Asembler:

```
pterm0: clr.w -(SP)
        trap #1
```

Funkcja ta powoduje zakończenie wykonywania procesu, zwolnienie zajmowanej pamięci i przekazanie sterowania do programu nadrzędnego lub systemu operacyjnego. Funkcja ta powinna być używana zawsze na końcu programu.

**1 Cconin**

C:

```
long Cconin( void );
```

Asembler:

```
cconin: move.w #1,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja oczekuje na naciśnięcie klawisza i zwraca (w rej. D0) jego kod, jednocześnie wyświetlając odpowiedni znak na ekranie. Młodsze słowo zwracanej danej zawiera kod ASCII naciśniętego klawisza (8 bitów), a starsze słowo kod klawiatury.

**2 Cconout**

C:

```
void Cconout( int c );
```

Asembler:

```
cconout: move.w #char,-(SP)
         move.w #2,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja wyświetla pojedynczy znak określony przez parametr char (kod ASCII) na ekranie w aktualnym miejscu kursora. Funkcja interpretuje prawidłowo kody sterujące emulatora terminala VT 52 (patrz artykuł wewnątrz numeru).

**3 Cauxin**

C:

```
int Cauxin( void );
```

Asembler:

```
cauxin: move.w #3,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja zwraca (w D0) jeden bajt odczytany ze złącza RS 232.

**4 Cauxout**

C:

```
void Cauxout( int byte );
```

Asembler:

```
cauxout: move.w #byte,-(SP)
         move.w #4,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja wysyła bajt (wartość byte) na złącze RS 232.

**5 Cprnout**

C:

```
int Cprnout( int char );
```

Asembler:

```
cprnout: move.w #char,-(SP)
         move.w #5,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja drukuje znak o kodzie określonym przez char na drukarce podłączonej do złącza Centronics. W wypadku gdy wydruk zakończył się pomyślnie w D0 zwracana jest wartość -1. Gdy przez 30 sekund drukarka nie odebrała wysłanego znaku zwracana jest wartość 0.

**6 Crawio**

C:

```
long Crawio( int char );
```

Asembler:

```
crawio: move.w char,-(SP)
        move.w #6,-(SP)
        trap #1
        addq.l #4,SP
```

Działanie tej funkcji zależy od wartości parametru char. Gdy char=255, to funkcja odczytuje i zwraca (w rejestrze D0) kod aktualnie naciśniętego klawisza, analogicznie jak funkcja cnecin. W wypadku, gdy w momencie wywołania funkcji nie był naciśnięty żaden klawisz, funkcja zwraca wartość 0. W wypadku gdy parametr char ma wartość różną od 255 traktowana jest on jako kod ASCII znaku, który zostaje wyświetlony na pozycji kursora. Funkcja reaguje poprawnie na kody kontrolne emulatora VT 52.

**7 Crawcin**

C:

```
long Crawcin( void );
```

Asembler:

```
crawcin: move.w #7,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja działa podobnie jak cconin, lecz wprowadzony znak nie jest wyświetlany.

**8 Cnecin**

C:

```
long Cnecin( void );
```

Asembler:

```
cnecin: move.w #8,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja działa identycznie jak crawcin.

**9 Cconws**

C:

```
int Cconws( const char *text );
```

Asembler:

```
cconws: move.l text,-(SP)
        move.w #9,-(SP)
        trap #1
        addq.l #6,SP
```

Funkcja wyświetla na ekranie na pozycji kursora tekst o adresie określonym przez parametr text. Ciąg znaków przeznaczonych do wyświetlenia musi być zakończony bajtem zerowym. Po wywołaniu funkcja zwraca ilość znaków jaka została wyświetlona (rej. D0). Funkcja ta poprawnie rozpoznaje kody kontrolne.

**10 Cconrs**

C:

```
void Cconrs( LINE *buf );
```

Asembler:

```
cconrs: move.l buf,-(SP)
        move.w #$0A,-(SP)
        trap #1
        addq.l #6,SP
```

Funkcja ta służy do wprowadzania ciągu znaków z klawiatury i jednoczesnego wyświetlania ich na ekranie. Jako parametr funkcja wymaga podania adresu bufora o specjalnej strukturze:

C:

```
typedef struct
{
    byte maxlen;
    byte actuallen;
    char buffer[255];
} LINE;
```

Asembler:

```
buf:
 maxlen: ds.b 1 ;maksymalna długość bufora
 actullen: ds.b 1 ;rzeczywista ilość odczytanych znaków
 buffer: ds.b 255 ;bufor
```

Pierwszy bajt bufora (parametr maxlen) określa spodziewaną (maksymalną) liczbę znaków które chcemy odczytać (czyli np.: 255). Drugi bajt (actuallen) zawiera po wywołaniu funkcji rzeczywistą liczbę wprowadzonych znaków. Pozostałe bajty bufora zawierają wprowadzone znaki. Wprowadzanie kończy się gdy naciśnięty zostanie klawisz RETURN lub zostanie wypełniony cały bufor. Funkcja ta interpretuje poprawnie jedynie niektóre kody sterujące.

**11 Cconis**

C:

```
int Cconis( void );
```

Asembler:

```
cconis: move.w #$0B,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja pozwala sprawdzić ile znaków znajduje się w buforze klawiatury. Po jej wywołaniu zwraca liczbę znaków lub wartość $FFFF (-1) gdy bufor jest pusty. Znaki zgromadzone w buforze można odczytać używając funkcji wprowadzania GEMDOS-u.

**14 Dsetdrv**

C:

```
long Dsetdrv( int drv );
```

Asembler:

```
dsetdrv: move.w #drv,-(SP)
         move.w #$0E,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja służy do ustawienia aktywnego napędu dysków. Parametr drv określa wybrany napęd: 0 — napęd A:, 1 — napęd B: itd. Po wywołaniu funkcja zwraca informację o dołączonych napędach. Bit 1 zwracanej danej (rejestr D0) odpowiada napędowi A:, bit 2 B: itd. Ustawienie bitu oznacza, że w systemie jest zainstalowana stacja o danym oznaczeniu.

### 16 Cconos

C:

```
int Cconos( void );
```

Asembler:

```
cconos: move.w #$10,-(SP)
        trap #1
        addq.l #2,SP
```

Ta funkcja zwraca status konsoli (w rej. D0). Jeżeli jest to wartość $FFFF można korzystać z funkcji wyprowadzania znaków, jeżeli inna wartość wyprowadzanie jest niemożliwe.

### 17 Cprnos

C:

```
int Cprnos( void );
```

Asembler:

```
cprnos: move.w #$11,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja pozwala stwierdzić gotowość drukarki podłączonej do złącza Centronics. Gdy drukarka jest w trybie ON LINE funkcja zwraca $FFFF (–1), w przeciwnym razie wartość 0.

### 18 Cauxis

C:

```
int Cauxis( void );
```

Asembler:

```
cauxis: move.w #$12,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja zwraca informację o stanie odbiornika złącza RS 232. Wartość $FFFF sygnalizuje, że w buforze znajduje się odebrany znak i można go odczytać przy pomocy odpowiedniej funkcji GEMDOS-u.

### 19 Cauxos

C:

```
int Cauxos( void );
```

Asembler:

```
cauxos: move.w #$13,-(SP)
        trap #1
        addq.l #2,SP
```

Funkcja zwraca wartość $FFFF gdy możliwe jest wysyłanie informacji przez złącze RS 232.

### 25 Dgetdrv

C:

```
int Dgetdrv( void );
```

Asembler:

```
dgetdrv: move.w #$19,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja podaje informację o aktualnie aktywnym napędzie dysków. Zwracana dana (rejestr D0) zawiera numer aktualnie aktywnego napędu: 0 — A:, 1 — B: itd.

### 26 Fsetdta

C:

```
void Fsetdta( DTA *buf );
```

Asembler:

```
fsetdta: move.l buf,-(SP)
         move.w #$1A,-(SP)
         trap #1
         addq.l #6,SP
```

Funkcja służy do ustalenia adresu 44 bajtowego bufora (wskazywanego przez parametr buf), niezbędnego systemowi do przeprowadzania niektórych operacji dyskowych. W normalnych warunkach adres bufora został określony przez system operacyjny i nie ma potrzeby jego zmiany. Struktura tego bufora jest następująca:

C:

```
typedef struct
{
    byte d_reserved [21];
    byte d_attrib;
    uword d_time;
    uword d_date;
    long d_length;
    char d_fname [14];
} DTA;
```

Asembler:

```
buf:
d_reserved: ds.b 21 ;21 bajtów zarezerwowanych
  d_attrib ds.b 1 ; bajt atrybutów pliku
  d_time ds.w 1 ; słowo zawierające czas utworzenia pliku
  d_date ds.w 1 ; słowo zawierające datę utworzenia pliku
  d_length ds.l 1 ; długie słowo określające długość pliku
  d_fname ds.b 14 ; 14 znaków nazwy pliku
```

### 32 Super

C:

```
long Super( void *stack );
```

Asembler:

```
super: move.l stack,-(SP)
       move.w #$20,-(SP)
       trap #1
       addq.l #6,SP
```

Działanie tej funkcji jest ściśle związane ze specyfiką procesorów MC 680x0. Funkcja ta pozwala przełączać MC 680x0 w tryb user mode i supervisor mode. W trybie user programista nie ma dostępu do rejestrów sprzętowych komputera i nie może używać pewnych rozkazów. W trybie supervisor istnieje dostęp do wszystkich rozkazów i rejestrów (patrz ST^Efan 3, 4/93 str. 27). Jako parametr stack podajemy wartość wskaźnika stosu, 0 gdy pozostawiamy poprzedni adres stosu lub 1 gdy chcemy sprawdzić w jakim trybie znajduje się procesor. Funkcja działa naprzemiennie, tzn. wywołanie w trybie user włącza tryb supervisor, zaś wywołanie w trybie supervisor włącza tryb user. Program użytkownika uruchamiany przez system operacyjny wywoływany jest zawsze w trybie user. Funkcja zmieniając adres stosu zwraca poprzednią jego wartość. Gdy jako parametr podana była wartość 1, funkcja zwraca wartość 0 jeżeli procesor jest w trybie user, lub 1 gdy pracuje w trybie supervisor.

### 42 Tgetdte

C:

```
unsigned int Tgetdate( void );
```

Asembler:

```
tgetdate:move.w #$2A,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja zwraca w aktualną datę ( w rejestrze D0). Format tej informacji jest następujący:

bity: 0–4 określają dzień miesiąca (1–31)

5–8 określają miesiąc (1–12)

9–15 określają rok, aby otrzymać właściwą datę należy do liczby wyrażonej tymi bitami należy dodać 1980.

### 43 Tsetdate

C:

```
unsigned int Tsetdate( unsigned int date );
```

Asembler:

```
tsetdate:move.w date,-(SP)
         move.w #$2B,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja ustawia aktualną datę. Wymaga podania parametru date w formacie j.w.

### 44 Tgettime

C:

```
unsigned int Tgettime( void );
```

Asembler:

```
tgettime:move.w #$2C,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja ta pozwala odczytać aktualny czas zegara GEMDOS-u. Zwracany jest aktualny czas w następującym formacie:

bity: 0–4 sekundy (przyrosty dwusekundowe)

5–10 minuty

11–15 godziny ( w systemie 24 godzinnym)

### 45 Tsettime

C:

```
unsigned int Tsettime( unsigned int time );
```

Asembler:

```
tsettime:move.w time,-(SP)
         move.w #$2D,-(SP)
         trap #1
         addq.l #4,SP
```

Funkcja ustawia zegar GEMDOS-u. Jako parametru wymaga podania aktualnego czasu w formacie j.w.

### 47 Fgetdta

C:

```
DTA *Fgetdta( void );
```

Asembler:

```
fgetdta: move.w #$2F,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja ta zwraca adres bufora dyskowego dla GEMDOS-u, ustawiany funkcją Fsetdta.

### 48 Sversion

C:

```
int Sversion( void );
```

Asembler:

```
sversion:move.w #$30,-(SP)
         trap #1
         addq.l #2,SP
```

Funkcja zwraca (w rejestrze D0) liczbę określającą numer wersji systemu operacyjnego:

1300 — GEMDOS 0.13 (TOS 1.00 i 1.02)

1500 — GEMDOS 0.15 (TOS 1.04 i 1.06)

1700 — GEMDOS 0.17 (TOS 1.062)

1900 — GEMDOS 0.19 (TOS 2.06 i 3.06)

3000 — GEMDOS 0.21 (?) (TOS 4.00)

### 49 Ptermres

C:

```
void Ptermres( long keepcnt, int retcode );
```

Asembler:

```
ptermres: move.w retcode,-(SP)
          move.l keepcnt,-(SP)
          move.w #$31,-(SP)
          trap #1
```

Funkcja ta, podobnie jak pterm0 ($00) służy do zakończenia wykonywania programu, ale dodatkowo pozwala przekazać kod zakończenia programu i chronić zarezerwowaną przez program pamięć. Jako parametr retcode przekazujemy zwykle wartość 0 gdy program wykonał się bezbłędnie lub 1 gdy wystąpił jakiś błąd. Parametr keepcnt określa jaki obszar pamięci (od początku programu) chcemy chronić przed zajęciem przez inne programy (obszar ten musi być zwiększony o 255 bajtów — długość strony bazowej (ang. basepage), zostanie to opisane w jednym z najbliższych ST^Efanów). Funkcja ta jest często używana do zakończenia programów rezydentnych.

**54 Dfree**

C:

```
int Dfree( DISKINFO *buffer,
int drive );
```

Asembler:

```
dfree: move.w drive,-(SP)
       move.l buffer,-(SP)
       move.w #$36,-(SP)
       trap #1
       addq.l #8,SP
```

Funkcja zwraca ilość wolnego miejsca na dysku. Parametr drive określa napęd:

0 — napęd aktualny

1 — napęd A:

2 — napęd B:

itd.

Parametr buffer jest adresem 16-bajtowego bufora, w którym umieszczone zostaną informacje o wolnym miejscu na dysku. Pierwsze długie słowo bufora zawiera liczbę wolnych zbitek (ang. cluster) na dysku. Drugie długie słowo określa całkowitą ilość zbitek na dysku. Trzecie długie słowo określa długość sektora w bajtach (zwykle w standardowych dyskietkach wynosi ona 512 bajtów). Ostatnie długie słowo zawiera liczbę sektorów tworzących jedna zbitkę (zwykle są to dwa sektory).

**57 Dcreate**

C:

```
int Dcreate(const char *path);
```

Asembler:

```
dcreate: move.l path,-(SP)
         move.w #$39,-(SP)
         trap #1
         addq.l #6,SP
```

Funkcja służy do zakładania katalogów. Parametr path zawiera adres ścieżki i nazwy katalogu, zakończonej bajtem zerowym. Po wywołaniu funkcji zwraca ona (rejestr D0) wartość 0 jeżeli operacja się powiodła, lub ujemny kod błędu.

**58 Ddelete**

C:

```
int Ddelete( const char *path
);
```

Asembler:

```
ddelete: move.l path,-(SP)
         move.w #$3A,-(SP)
         trap #1
         addq.l #6,SP
```

Funkcja służy do usuwania katalogów, parametry funkcji podawane są jak wyżej.

**59 Dsetpath**

C:

```
int Dsetpath( const char *path
);
```

Asembler:

```
dsetpath: move.l path,-(SP)
          move.w #$3B,-(SP)
          trap #1
          addq.l #6,SP
```

Funkcja zmienia aktualną ścieżkę dostępu. Parametr path zawiera adres nowej ścieżki zakończony bajtem zerowym.

**60 Fcreate**

C:

```
long Fcreate( const char *file-
name, int attr );
```

Asembler:

```
fcreate: move.w attr,-(SP)
         move.l filename,-(SP)
         move.w #$3C,-(SP)
         trap #1
         addq.l #8,SP
```

Funkcja służy do zakładania nowego pliku. Istnienie tej funkcji spowodowane jest specyfiką systemu GEMDOS. Zanim zaczniemy przeprowadzać jakikolwiek operacje na pliku najpierw musimy go otworzyć specjalną funkcją, a po zakończeniu operacji należy go zamknąć. Taki sposób postępowania jest właściwy jedynie w wypadku plików już istniejących, zaś gdy chodzi o nowe pliki do ich zakładania służy funkcja fcreate. Parametr filename określa adres nazwy zakładanego pliku (zakończonej bajtem zerowym). Parametr attr określa atrybuty pliku. Oto znaczenie poszczególnych wartości parametru attr.

$00 — plik do zapisu i odczytu

$01 — plik tylko do odczytu

$02 — plik ukryty

$04 — plik systemowy

$08 — nazwa dysku

$10 — plik jest podkatalogiem

$20 — plik został zamknięty

Oczywiście poszczególne atrybuty można łączyć, dodając do siebie odpowiednie wartości.

Po wywołaniu funkcja zwraca w (rejestrze D0) numer łącznika pliku lub ujemny kod błędu. Łącznik pliku jest jego identyfikatorem używanym przy dalszych operacjach, a więc odczycie, zapisie i zamykaniu. Po użyciu funkcji fcreate dostęp do pliku jest możliwy bez dodatkowego otwierania funkcją fopen.

**61 Fopen**

C:

```
long Fopen( const char *file,
int mode );
```

Asembler:

```
fopen: move.w mode,-(SP)
       move.l file,-(SP)
       move.w #$3D,-(SP)
       trap #1
       addq.l #8,SP
```

Funkcja ta służy do otwierania plików. Plik można otworzyć w trybie do odczytu, zapisu lub odczytu i zapisu, decyduje o tym wartość parametru mode:

0 — odczyt

1 — zapis

2 — zapis/odczyt

Parametr file określa adres nazwy pliku, zakończonej bajtem zerowym. Funkcja po wywołaniu zwraca łącznik pliku, analogicznie jak funkcja fcreate. W systemie GEMDOS dostęp do plików poprzez funkcje fread ifwrite jest sekwencyjny. Przy otwieraniu tworzony jest wskaźnik do danych, który określa ostatni czytany lub zapisany bajt. Przy każdym odczycie/zapisie wskaźnik do pliku jest odpowiednio zwiększany. Dostęp do danych przy użyciu fread/fwrite jest możliwy jedynie od aktualnej pozycji wskaźnika pliku do końca zbioru.

**62 Fclose**

C:

```
int Fclose( int handle );
```

Asembler:

```
fclose: move.w handle,-(SP)
        move.w #$3E,-(SP)
        trap #1
        addq.l #4,SP
```

Funkcja zamyka plik, jako parametr handle należy podać numer łącznika.

**63 Fread**

C:

```
long Fread( int handle, long
count, void *buffer );
```

Asembler:

```
fread: move.l buffer,-(SP)
       move.l count,-(SP)
       move.w handle,-(SP)
       move.w #$3F,-(SP)
       trap #1
lea 12(SP),SP
```

Funkcja pozwala na odczyt pliku. Parametr buffer określa adres pod który będą przesyłane odczytane dane. Parametr handle zawiera numer łącznika, a więc określa który plik będzie odczytywany. Parametr count określa ilość bajtów, które chcemy odczytać. W wypadku gdy nie znamy długości pliku jako wartość lenght podajemy $FFFFFFFF. Plik jest wtedy czytany do napotkania ostatniej zbitki zajętej przez jego dane (jest to więc trochę więcej niż wynosi długość pliku na dysku!!!), co w niczym nie przeszkadza. Po wywołaniu funkcji zwraca ona liczbę odczytanych bajtów, lub ujemny kod błędu.

**64 Fwrite**

C:

```
long Fwrite( int handle, long
count, void *buffer );
```

Asembler:

```
fwrite: move.l buffer,-(SP)
        move.l count,-(SP)
        move.w handle,-(SP)
        move.w #$40,-(SP)
        trap #1
        lea 12(SP),SP
```

Funkcja służy do zapisu pliku. Parametry wywołania są analogiczne jak dla funkcji fread, z tym, że parametr lenght musi być dokładnie określony.

**65 Fdelete**

C:

```
int Fdelete( const char *na-
me );
```

Asembler:

```
fdelete: move.l name,-(SP)
         move.w #$41,-(SP)
         trap #1
         addq.l #6,SP
```

Funkcja służy do kasowania plików. Jako parametr name należy podać adres pełnej nazwy pliku, lub ścieżki dostępu zakończonej bajtem zerowym.

**66 Fseek**

C:

```
long Fseek( long offset, int
handle, int seekmode );
```

Asembler:

```
fseek: move.w seekmode,-(SP)
       move.w handle,-(SP)
       move.l offset,-(SP)
       move.w #$42,-(SP)
       trap #1
       lea 10(SP),SP
```

Funkcja pozwala uzyskać dostęp do konkretnych fragmentów pliku poprzez modyfikację wskaźnika pamiętającego ostatnio odczytany/zapisany bajt. Parametr offset podaje o ile chcemy przesunąć wskaźnik pliku, handle stanowi identyfikator pliku, zaś parametr seekmode określa sposób przesunięcia. Dla seekmode=0 wskaźnik pliku jest zerowany i offset stanowi żądane przesunięcie od początku pliku (wartość tylko dodatnia). Dla seekmode=1 wskaźnik jest przesuwany o odpowiednią wartość (dodatnią lub ujemną) od swojej aktualnej pozycji. Dla seekmode 2 wskaźnik jest ustawiany na koniec pliku i cofany o odpowiednią liczbę bajtów (ujemną).

**67 Fattrib**

C:

```
int Fattrib( const char *file-
name, int wflag, int attr );
```

Asembler:

```
fattrib: move.w attr,-(SP)
```

```
         move.w wflag,-(SP)
         move.l filename,-(SP)
move.w #$43,-(SP)
trap #1
lea 10(SP),SP
```
Funkcja pozwala odczytać lub zmienić wartość atrybutu pliku. Parametr filename stanowi adres nazwy pliku zakończonej bajtem zerowym, parametr wflag określa czy chodzi o odczyt atrybutów (wflag=0) czy o jego zmianę (wflag=1). Parametr attr określa właściwe atrybuty pliku — patrz opis funkcji Fcreate.

**69 Fdup**

C:

```
long Fdup( int handle );
```

Asembler:

```
fdup: move.w handle,-(SP)
      move.w #$45,-(SP)
      trap #1
      addq.l #4,SP
```
Funkcja pozwala stworzyć kopię dla jednego z kanałów wejścia wyjścia. Parametr handle zawiera numer kanału którego identyfikator chcemy uzyskać. Pozwala to np. na zapis do RS 232 funkcją Fwrite. Numery kanałów zostały w następujący sposób przyporządkowane poszczególnym urządzeniom.

kanał: 0 — CON: (wejście konsoli — klawiatura)

1 — CON: (wyjście konsoli — ekran)

2 — AUX: (złącze RS 232)

3 — PRN: (złącze Centronics)

**70 Fforce**

C:

```
int Fforce( int stch, int nonstdh );
```

Asembler:

```
fforce: move.w stch,-(SP)
        move.w nonstdh,-(SP)
        move.w #$46,-(SP)
        trap #1
        addq.l #6,SP
```
Funkcja pozwala przypisać numerowi kanału jednego z urządzeń (np. Centronics) identyfikator pliku. Pozwala to, używając funkcji wysyłających dane do tego kanału skierować je do pliku.

**71 Dgetpath**

C:

```
int Dgetpath( char *path, int driveno );
```

Asembler:

```
dgetpath: move.w driveno,-(SP)
          move.l path,-(SP)
          move.w #$47,-(SP)
          trap #1
          addq.l #8,SP
```
Funkcja pobiera aktualną ścieżkę dostępu do aktualnego katalogu. Parametr driveno określa napęd (0 — aktualnie wybrany, 1 — A:, 2 — B: itd.), zaś path stanowi adres 64 bajtowego bufora w którym ścieżka ta zostanie zapamiętana,

**72 Malloc**

C:

```
void *Malloc( long number );
```

Asembler:

```
malloc: move.l number,-(SP)
        move.w #$48,-(SP)
        trap #1
        addq.l #6,SP
```
Funkcja pozwala zarezerwować określony obszar pamięci. Żądaną ilość pamięci należy podać jako parametr number. Funkcja zwraca adres zarezerwowanego obszaru lub −1 gdy wystąpił błąd. Podanie jako parametru number wartości −1 powoduje, że funkcja zwraca ilość dostępnej pamięci.

**73 Mfree**

C:

```
int Mfree( void *block );
```

Asembler:

```
mfree: move.l block,-(SP)
       move.w #$49,-(SP)
       trap #1
       addq.l #6,SP
```
Funkcja pozwala zwolnić zarezerwowaną pamięć. Parametr block określa adres zarezerwowanego obszaru.

**74 Mshrink**

C:

```
int Mshrink(0, void *block, long newsiz );
```

Asembler:

```
mshrink: move.l newsiz,-(SP)
         move.l block,-(SP)
         pea $4A0000
         trap #1
         lea 12(SP),SP
```
Funkcja pozwala zmienić rozmiar zarezerwowanego obszaru pamięci. Parametr block określa adres zarezerwowanego obszaru, a parametr newsiz żądany nowy rozmiar tego obszaru.

Podczas uruchamiania programu przez GEMDOS i przydzielana jest mu cała dostępna aktualnie pamięć. Często jest to niekorzystne, gdyż uniemożliwia nakładkowanie innych programów. W związku z tym programista powinien na początku programu określić ilość potrzebnej pamięci i zmienić jej przydział używając funkcji Mshrink.

**75 Pexec**

C:

```
long Pexec( int mode, char *ptr1, void *ptr2, void *ptr3 );
```

Asembler:

```
pexec: move.l ptr3,-(SP)
       move.l ptr2,-(SP)
       move.l ptr1,-(SP)
       move.w mode,-(SP)
       move.w #$4B,-(SP)
       trap #1
       lea 16(SP),SP
```
Funkcja ta pozwala nakładkować programy, czyli uruchamiać jedne programy z wnętrza innych. Twórcy systemu przewidzieli możliwość podania każdemu programowi tzw. otoczenia (ang. enviroment), czyli informacji o dostępnej pamięci itp. oraz kilka sposobów uruchomienia programu — natychmiastowe, tylko wczytanie, relokację kodu, uruchomienie programu wcześniej wczytanego i zwolnienie pamięci po zakończonym programie. Niestety w systemie GEMDOS do wersji 0.15 (do TOS-u 1.06) istnieją pewne błędy, które powodują, że nie możliwe jest wykorzystanie tych możliwości. Pewnie działa najprostszy przypadek — wczytanie i uruchomienie programu bez podawania otoczenia. Parametr mode i ptr3 powinny być równe 0, parametr ptr2 wskazuje adres linii poleceń zakończonej bajtem zerowym (jak dla programów TTP), zaś parametr ptr1 wskazuje nazwę pliku zakończoną bajtem zerowym. Po zakończeniu działania tak uruchomionego programu funkcja zwraca kod powrotu retcode (patrz funkcja Pterm). Więcej szczegółów o działaniu funkcji Pexec znajdziecie w odrębnym artykule w następnym numerze.

**76 Pterm**

C:

```
void Pterm( int retcode );
```

Asembler:

```
pterm: move.w retcode,-(SP)
       move.w #$4C,-(SP)
       trap #1
```
Funkcja kończy wykonywanie programu, zwalania zajętą pamięć, i wraca do programu nadrzędnego, lub systemu operacyjnego przekazując kod powrotu retcode.

**78 Fsfirst**

C:

```
int Fsfirst( const char *filename, int attr );
```

Asembler:

```
fsfirst: move.w attr,-(SP)
         move.l filename,-(SP)
         move.w #$4E,-(SP)
         trap #1
         addq.l #8,SP
```
Funkcja pozwala odszukać w katalogu plik i umieścić informacje o nim w buforze GEMDOS-u (patrz Setdta). Parametrami są adres nazwy pliku (można użyć znaków ? i *) określony przez filename oraz atrybut szukanego pliku (attr) pozwalający szukać plików zwykłych (attr=0), systemowych (attr=4) czy podkatalogów (attr=10).

**79 Fsnext**

C:

```
int Fsnext( void );
```

Asembler:

```
fsnext: move.w #$4F,-(SP)
        trap #1
        addq.l #2,SP
```
Funkcja odczytuje informacje o kolejnym pliku spełniającym warunki określone przez Fsfirst.

**86 Frename**

C:

```
int Frename(0, const char *oldname, const char *newname );
```

Asembler:

```
frename: move.l newname,-(SP)
         move.l oldname,-(SP)
         pea $560000
         trap #1
         lea 12(SP),SP
```
Funkcja pozwala zmienić nazwę pliku. Parametry oldname i newname są wskaźnikami do starej i nowej nazwy pliku.

**87 Fdatime**

C:

```
int Fdatime( DOSTIME *timeptr, int handle, int wflag );
```

Asembler:

```
fdatime: move.w wflag,-(SP)
         move.w handle,-(SP)
         move.w timeptr,-(SP)
         move.w #$57,-(SP)
         trap #1
         lea 10(SP),SP
```
Funkcja pozwala odczytać lub zmienić datę założenia pliku na dysku. Parametr wflag określa czy chodzi o odczyt (wartość 1) czy o zapis (wartość 0). Parametr handle jest identyfikatorem pliku, zaś timeptr wskazuje 4 bajtowy bufor zawierający czas i datę w formacie używanym przez funkcje Tgettime i Tgetdate.

To już koniec ostatniego artykułu z serii „TOS od środka". Mam nadzieję, że to czysto teoretyczne przedstawienie funkcji systemu TOS nie zniechęciło zbyt wielu czytelników. W następnym ST'fanie postaram się przedstawić w jaki sposób używać funkcji systemu we własnych programach w asemblerze.

**Literatura:**

[1] ST-STE-TT Profibuch (Jankowski, Rabich, Reschke — Sybex Verlag).

Tomasz Talkowski

# Lynx do kieszeni

Mały, elegancki, pasujący do dłoni — nowy komputer, a właściwie konsola do gier firmy Atari — Lynx.

Lynx ma wbudowany joypad, dwa przyciski fire (podwójne), kolorowy wyświetlacz ciekłokrystaliczny, wyjście słuchawkowe (stereo) oraz gniazdko zasilania i Comlynx, pozwalające na połączenie ze sobą do sześciu Lynxów. Obudowa jest estetyczna, a wyświetlacz umieszczony pod szybką, co chroni go przed uderzeniami. Jest to w końcu urządzonko przenośne — przeznaczone do wyżywania się w czasie podróży lub nudnych lekcji. Przydaje się także w czasie dłuższych seriali-tasiemców do denerwowania zebranej przed telewizorem rodziny odgłosami z wcale nie cichego, wbudowanego, głośniczka.

Sercem Lynxa są dwie kości: Mikey i Suzy. Mikey zawiera mikroprocesor 65C02, układ dźwiękowy, układ podobny do Shiftera w ST — sprzęg dla LCD, system timerów, kontroler przerwań oraz UART (Universal Asynchronous Receiver-Tranmitter) — zajmujący się komunikacją w sieci. Procesor 65C02 (8-bitowy) w Lynxie jest w pełni zgodny z tymi w Atari XL/XE, Apple II. Posiada dodatkowe tryby adresowania oraz nowe instrukcje. Jego budowa podobna jest do tej z procesorów typu RISC.

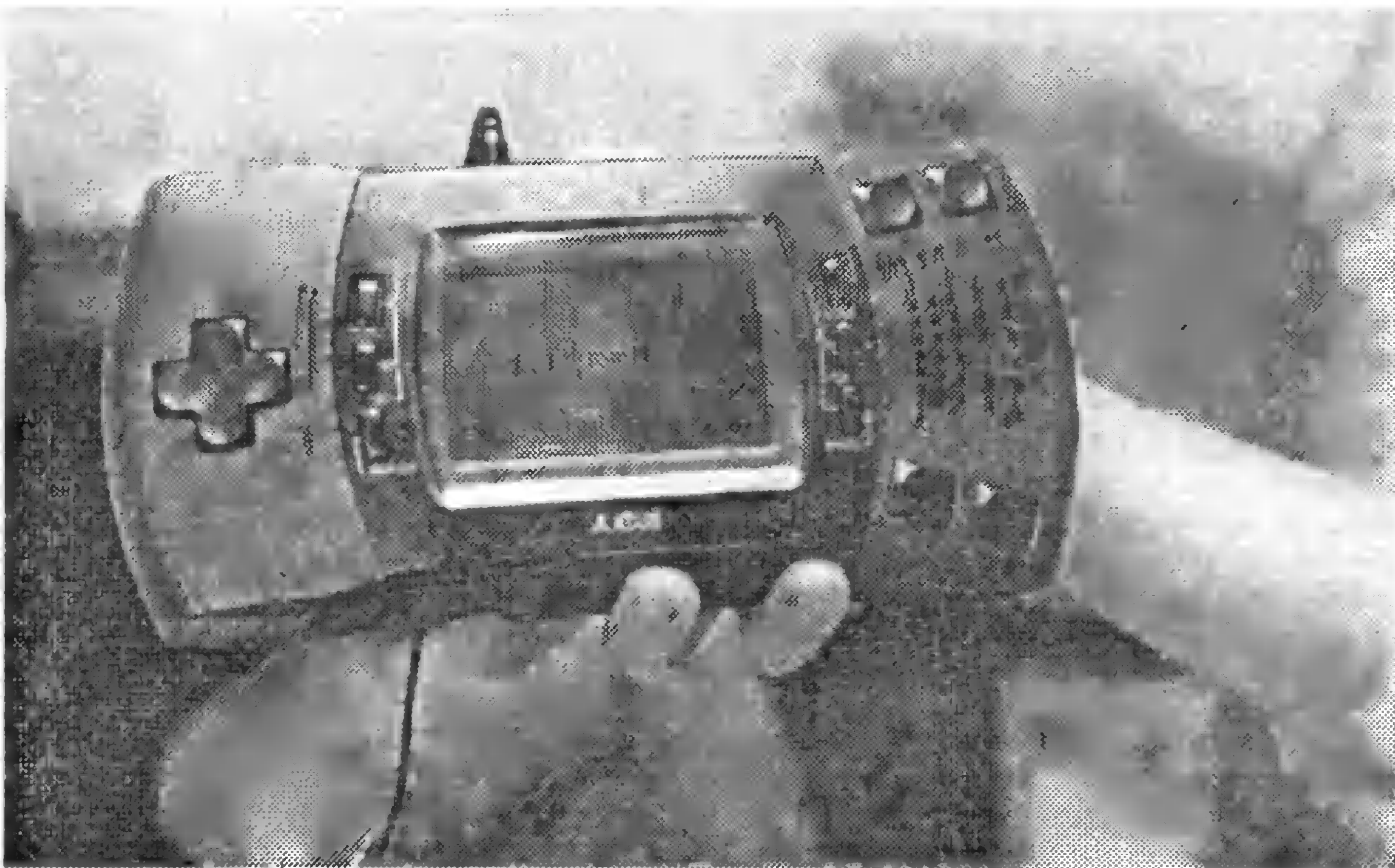
Prędkość zegara głównego wynosi 16 MHz. Sygnał ten dzielony jest przez cztery aby wytworzyć takt dla procesora 65C02 czyli 4 MHz (dla porównania: Atari XL/XE w wersji PAL ma zegar 1,72 MHz).

Ekran Lynxa — LCD — potrafi wyświetlić w poziomie 160 a w pionie 102 triady. Każda triada składa się z czerwonej, zielonej i niebieskiej plamki, mogącej wyświetlać 16 poziomów nasycenia koloru. W sumie na ekranie może zostać wyświetlone 4096 różnych kolorów. Każda triada reprezentowana jest przez cztery bity video RAM. Pozwala to na uzyskanie 16 kolorów jednocześnie na ekranie (podobnie jak ma to miejsce w niskiej rozdzielczości Atari ST[E]). Wirtualne bufory mogą być wykorzystywane w różnych funkcjach graficznych takich jak wykrywanie kolizji, *double buffering*, itp. Mikey — układ tłumaczący video RAM ma sygnały zrozumiałe dla LCD taktowany jest bezpośrednio z zegara głównego czyli z częstotliwością 16 MHz. Lynx posiada bardzo szybki blitter (Suzy), pozwalający na uzyskanie szybkich duszków, skalowanie, rozciąganie i cieniowanie. Suzy może skalować duszki w skali 1/256 czyli duszek może być maksymalnie 256 razy większy lub 256 razy mniejszy od oryginału. Funkcje cieniowania i rozciągania czynią możliwym pokrywanie dużych powierzchni ekranu używając bardzo małego duszka. Duszek zdefiniowany jako pojedynczy punkt (4 bity) może być z łatwością rozciągnięty w linię lub w kolorowe pole (zależnie od użytej funkcji). Dane dotyczące duszków mogą być kompresowane na bieżąco, więc nie muszą zajmować miejsca w pamięci.

Elektronika zajmująca się dźwiękiem znajduje się w kości Mikey — osiem generatorów podobnych do tych w Atari XL/XE oraz z dwa przetworniki.

Lynx kosztuje w Polsce 2,7 mln zł — myślę, że jest to ciekawa cena.

❑



Armara

# Mat Puzzle 3xTAK

Edukację szkolną zakoczyłem już spory czas temu, dlatego na wszelkie programy edukacyjne spoglądam krytycznie i z dystansem. Po kilku latach łakomego „połykania" programów ułatwiających naukę trudno mi jest oprzeć się wrażeniu, że polski rynek edukacyjnego oprogramowania edukacyjnego został zalany tanią tandetą. Z tym większą zatem przyjemnością (a także z pewną dozą nieśmiałości) sięgnąłem po mało znany program edukacyjny Mat Puzzle. Zgodnie z tytułem recenzji, z całą odpowiedzialnością mogę ten właśnie program polecić wszystkim uczącym się matematyki. A dlaczego?

**1 x TAK**

Z matematyką jest jak z „kucharzeniem": albo ma się w sobie określone zdolności — i wtedy wszystko jest łatwe i przyjemne — albo tych zdolności się nie ma — a wtedy nauka matematyki jest drogą przez mękę. Program Mat Puzzle przeznaczony jest zarówno dla tych z matematycznymi zdolnościami jak i dla tych, dla których matematyka jest najkoszmarniejszym snem. Pierwszym bowiem pozwala wygimnastykować mózg w zakresie wykonywania czterech podstawowych działań matematycznych w zbiorze liczb całkowitych od 1 do 1000; drugim zaś pozwala w sposób całkiem bezbolesny (ba! nawet bardzo przyjemny...) zawrzeć — być może trwałą — przyjaźń z matematyką.

**2 x TAK**

Nawet najbardziej znakomity program edukacyjny może się nam znudzić, jeśli wraz z porcją wiedzy nie otrzymamy wystarczającej ilości rozrywki. Autor Mat Puzzli zadbał, by użytkownikom jego programu naprawdę się nie nudziło. Zajęcie jest naprawdę wciągające, choć nie wszyscy przepadają za układankami. Chyba jednak nie ma człowieka, który nie lubiłby Kaczora Donalda i innych postaci z bajek (a to one są właśnie treścią układanych obrazków).

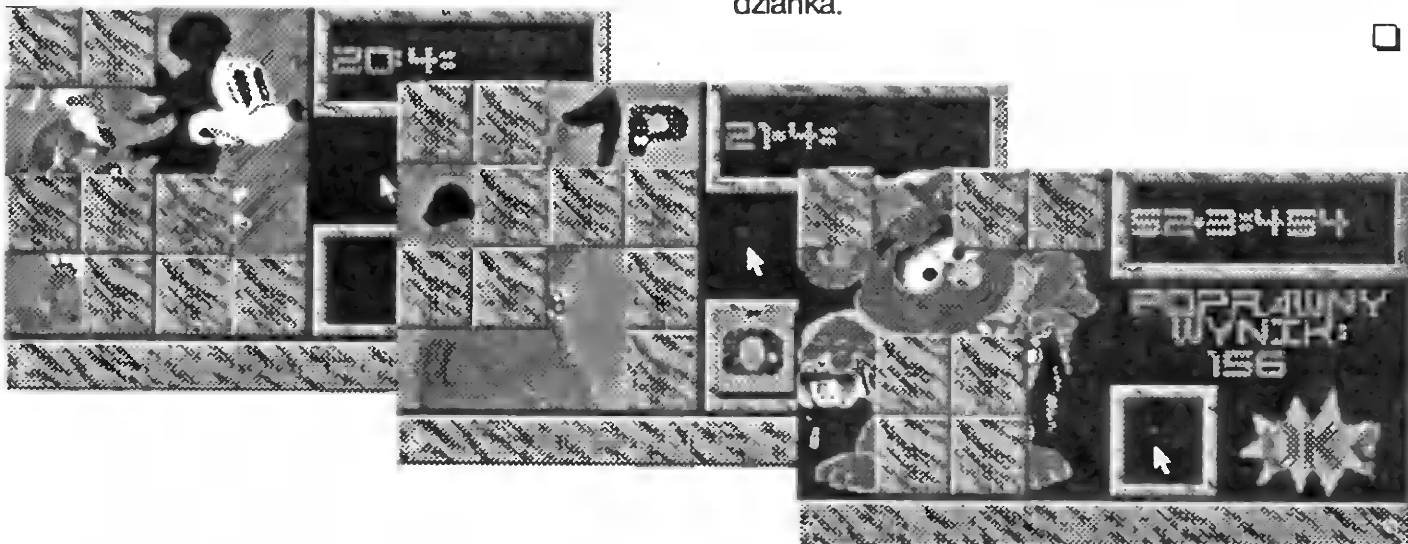
**3 x TAK**

Z wieloma programami komputerowymi jest tak, że sięgają po nie użytkownicy z określonych grup wiekowych. Tym czasem Mat Puzzle, choć to program edukacyjny — nie rozczaruje siedmiolatka, ani wiekowego znawcy przedmiotu. Program ma także swoje wady, na przykład jak na moją cierpliwość za długo się ładuje. Przydałyby się bardziej skomplikowane działania matematyczne. Ale te drobne niedociągnięcia są niemal niewidoczne w obrazie całości. Polecam zatem wszystkim kupno oryginalnej kopii tego programu. Odradzam jednocześnie poszukiwania Mat Puzzli na giełdach komputerowych — krąży na nich tylko wersja demonstracyjna programu.

Oczywiście trudno jest zaspokoić oczekiwania wszystkich (zwłaszcza wybrednych) użytkowników oprogramowania. Dlatego uczmy się matematyki, układajmy puzzle i miejmy nadzieję, że autorzy programu wypuszczą wkrótce jego nową poszerzoną wersję.

Uwaga! Wiadomość z ostatniej chwili. Trwają prace nad kontynuacją programu. W kolejnej części dzieci będą uczyły się działań na ułamkach, a także pisemnego wykonywania czterech podstawowych działań. Tym razem (zamiast układać puzzle) będziemy się bawili... a to już niespodzianka.

❑





Tomasz Dajczak

# Sound Merlin

**Pewnego dnia postanowiłem kupić sobie sampler. Ponieważ moim hobby jest muzyka, przeto uznałem to urządzenie za dość przydatne.**

Nie miałem ochoty jechać do większego miasta, więc sampler zamówiłem wysyłkowo u producenta (w firmie TOMS). Ponieważ producent obiecywał, że przetwornik będzie działał z każdym oprogramowaniem wykorzystującym port równoległy, nie zamówiłem polecanego przez firmę programu Eter (co będę przepłacał za jakiś program grube pieniądze, jeśli mogę kupić podobny za półdarmo na giełdzie? — pomyślałem i niedługo prawie pożałowałem swego skąpstwa).

Sampler przyszedł pocztą po kilku dniach. Po otwarciu paczki ujrzałem przetwornik, podłączyłem go do komputera i... okazało się, że żaden z moich programów nie współpracuje z samplerem podłączanym do portu równoległego. Całe szczęście, że po kilku dniach stałem się posiadaczem programu Sound Merlin.

Sound Merlin jest programem shareware, wydanym przez niemiecką firmę Tommy Software (znaną między innymi ze znakomitego programu graficznego Mega Paint). Jest to więc jeden z nielicznych programów, jakie można mieć całkiem legalnie, bez zbytnich nakładów finansowych.

Merlin służy głównie do digitalizowania dźwięku, ale można za jego pomocą robić również wiele innych rzeczy. Wbudowany sequencer i automat perkusyjny powodują, że Merlin to trzy programy w jednym.

Jego obsługa jest według mnie łatwa, prosta i przyjemna. Ekran jest podzielony na kilka części spełniających określone funkcje.

Po prawej stronie znajdują się stale widoczne piktogramy, umożliwiające zmianę kolorów ekranu, włączenie przyspieszacza myszki, ustawienie parametrów programu oraz wyprowadzenie dźwięku na przetwornik cyfrowo-analogowy (zewnętrzny). Pozwala to na znaczne poprawienie jakości dźwięku, ale ma jedną, podstawową wadę. Przetwornik D/A, podobnie jak sampler używa złącza centronics więc nie można używać ich jednocześnie.

Po lewej stronie ekranu znajduje się obrazkowe menu umożliwiające uruchomienie akcesoriów, zakończenie programu oraz wybieranie aktywnego „modułu" (sampler, edytor sampli, automat perkusyjny itp.). Słowo moduł napisałem w cudzysłowie, ponieważ nie chodzi tu o moduły w pełnym tego słowa znaczeniu (jak w Calamusie SL) i nie można ich doładowywać osobno ani zastępować innymi.

W górnej części ekranu widzimy powiększony obraz obrabianego sampla oraz ilość wolnej pamięci. Okienko to może spełniać również np. rolę oscyloskopu czy equalizera.

W prawym dolnym rogu, widzimy okienko, w którym mogą znajdować się parametry sampla, automatu perkusyjnego, systemu lub lista próbek znajdujących się w pamięci (praktycznie można załadować ich tyle, na ile pamięć pozwala). Zawartość okienka zmieniamy tupiąc myszką nad jego nazwą. Przy pomocy jednego z menu umieszczonych w tym miejscu można również sformatować dyskietkę.

Na dole, w środkowej części ekranu znajduje się najważniejsze menu, przy pomocy którego uruchamiamy poszczególne opcje programu. Jego zawartość się zmienia — zależnie od aktywnego modułu. W większości jednak możemy zauważyć ikony Play Actual, Play Window i Play Window Hold służące do odtwarzania dźwięku. Często spotkać można również strzałkę z podpisem Previous Menu służącą do powrotu do poprzedniego zestawu piktogramów.

Najważniejszym modułem programu jest SAMPLER — umożliwiający ustawienie parametrów próbkowania oraz przeprowadzenie digitalizacji. Częstotliwość można regulować bardzo dokładnie, aż do 50 kHz. Interesującymi opcjami są Test Record Level (podgląd poziomu nagrania), Oscyloskop oraz Monitor (podsłuch dźwięku z magnetofonu przez głośnik monitora, bardzo praktyczne przy ocenie jakości digitalizacji „na słuch"). Można wywołać również bardzo efektownie wyglądający equalizer. Po zdigitalizowaniu można zapętlić sampla oraz nadać mu nazwę.

Po wczytaniu dźwięku do pamięci komputera, należy wyciąć niepotrzebne przerwy na początku i końcu sampla. Do tego służy moduł „Manipulate". Wycinanie fragmentów jest zrobione bardzo logicznie — można usunąć zarówno wybraną część sampla jak i marginesy. Program posiada pokaźną bibliotekę filtrów cyfrowych — dolno- i górnoprzepustowych oraz pasmowych, można miksować sample ze sobą, łączyć je itp. Można również przesunąć wykres sampla w górę lub dół albo odwrócić go tyłem. Brakuje tylko efektów echa i pogłosu znanych z programu STOS Maestro. Funkcja Undo pozwala na bezpieczne eksperymentowanie nad dźwiękiem. Przydatna jest również konwersja sampli z jednej częstotliwości na inną, na przykład gdy nagle zacznie kończyć się pamięć.

Jeśli chcemy bardzo dokładnie ustawić zakres sampla, uruchamiamy moduł Fine Edit. Dzięki niemu można obejrzeć naszą próbkę w bardzo dużym powiększeniu.

Przy pomocy modułu Keyboard przyporządkowujemy sample poszczególnym klawiszom komputera. Na pierwszy rzut oka może to trochę dziwić, ale po głębszym namyśle można znaleźć parę zastosowań dla tej opcji. Choćby udźwiękowienie szkolnego przedstawienia (ryk smoka, odgłosy walki, dzwony). W wielu przypadkach jest to praktyczniejsze od magnetofonu — odpadają kłopoty z przewijaniem taśmy oraz poszukiwaniem w ostatniej chwili potrzebnego efektu. Rozkład klawiatury można zapisać na dysk. Przy odczycie, potrzebne sample są doładowywane automatycznie (o ile nie ma ich w pamięci). Jakość dźwięku jest wystarczająca (zwłaszcza jeśli puścimy dźwięk przez zewnętrzny przetwornik D/A i wzmacniacz).

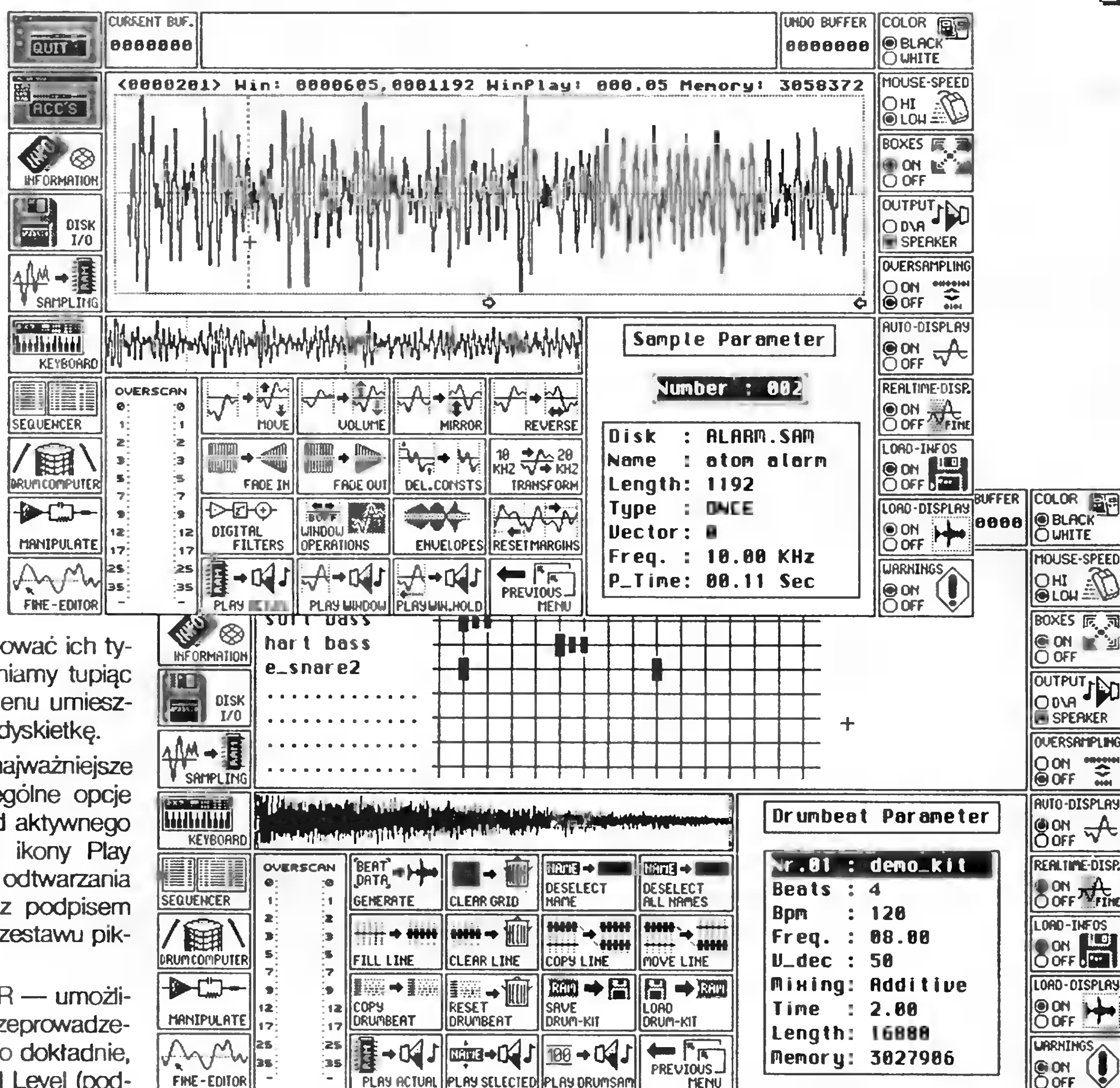
Sequencer mimo swej nazwy nie ma nic wspólnego z MIDI. Za jego pomocą ustalamy w jakiej kolejności mają być odtwarzane sample znajdujące się w pamięci, oraz jakie między nimi mają być odstępy. Jedyne zastosowanie, jakie przyszło mi do głowy to udźwiękowienie amatorskiego filmu video czy nagrania magnetofonowego.

Najbardziej zaskoczył mnie automat perkusyjny. Perkusja w edytorze sampli? — zapytacie. Dlaczego nie? Mamy do dyspozycji osiem śladów, sampli tyle ile się zmieści w pamięci. Wystarczy przyporządkować poszczególnym ścieżkom nazwy sampli i umieścić dźwięki w odpowiednich miejscach. Poprzez kilkukrotne puknięcie myszą można regulować głośność instrumentu. Aby wysłuchać dzieła należy wybrać Generate i poczekać kilka sekund. Zostanie wygenerowany nowy sampling zawierający kompletny rytm.

Moduł operacji dyskowych pozwala na zapis i odczyt sampli w kilku różnych formatach, tak więc wykorzystanie digitalizacji we własnych programach nie powinno nikomu sprawić kłopotu (tym bardziej, że odtwarzanie sampli było opisywane w ST$^E$fanie). Proszę jednak o zwrócenie uwagi na jeden istotny szczegół. Posiadana przeze mnie wersja programu źle nagrywa pierwszy sampling (po wyjściu z programu okazuje się, że ma on zero bajtów długości). Radzę więc dokonanie pierwszej operacji zapisu dwa razy (najlepiej pod różnymi nazwami).

Mimo przedstawionych przeze mnie (nielicznych) wad i niedociągnięć, Merlin jest bardzo dobrym programem i mogę polecić go każdemu z czystym sumieniem. Wymaga on tylko pół megabajty pamięci oraz działa również w średniej rozdzielczości, z czego często korzystam, bo głośniczek w monitorze SM 124 jest nienajlepszy. Pozostaje mi tylko życzyć Wam udanych digitalizacji i gumowej pamięci.

❑



# Program za grosik

**Uwaga, uwaga. Wszystkich Czytelników zamawiających programy Public Domain z naszej rubryki „Program za grosik" pragnę poinformować, że zaszły znaczące zmiany. Spowodowane są one ubytkami w naszej bibliotece PD, o czym niektórzy mogli się przekonać osobiście (niestety). Zmuszeni byliśmy uporządkować nieco nasz zbiór, a przy okazji zmienić jego organizację.**

Na początek parę wyjaśnień:

Programy PD podzielić można na trzy podstawowe grupy:

— Programy Public Domain — są to programy, których autorzy zrzekli się prawa autorskiego i przekazali je do swobodnego rozpowszechniania. Programy te nie posiadają instrukcji (w postaci książeczki), a za ich działanie autorzy nie ponoszą odpowiedzialności. Programy te mogą być modyfikowane, kopiowane, itp.

— Programy Freeware — są to programy, które wolno swobodnie kopiować i rozpowszechniać, ale posiadają zastrzeżone prawa autorskie i nie wolno ich modyfikować.

— Programy Shareware — są to programy, które można swobodnie rozpowszechniać, lecz gdy się chce z nich stale korzystać należy uiścić autorowi określoną przez niego opłatę licencyjną. Zwykle otrzymuje się wtedy pełną dokumentację, najnowszą wersję programu i możliwość wymiany na następną.

Należy pamiętać, że programy PD są jedynymi programami, które można legalnie kopiować w państwach o normalnym prawie autorskim. Biblioteki PD mają prawo pobierać opłaty w wysokości pokrywającej koszt dyskietek, obsługi klienta i pozwalającej utrzymać działalność biblioteki. Opłata ta nie jest równoznaczna z opłatą licencyjną za korzystanie z programów shareware (powinno się ją uiszczać bezpośrednio autorowi).

Nasz zbiór PD podzieliliśmy tematycznie na:

— programy narzędziowe (pakery, kopiery, ramdyski, programy formatujące, itp.) oznaczone literą N i kolejnym numerem dysku,

— programy edukacyjne (służące do nauki czegokolwiek) oznaczone literą E i numerem dysku,

— programy użytkowe (edytory, arkusze kalkulacyjne, itp.) oznaczone litera U i numerem dysku,

— programy graficzne (a także np.: zbiory grafik, czy fontów) oznaczone literą G,

— programy muzyczne (programy MIDI, trackery, obróbka sampli, moduły) oznaczone literą M i numerem dysku,

— gry na monitor kolorowy, oznaczone literami GK i numerem,

— gry na monitor monochromatyczny oznaczone literami GM i numerem,

— programy demonstracyjne i pokazowe oznaczone literą D i numerem,

— programy na Falcona oznaczone literą F oraz jedną kombinacją z powyższych (dyskietki HD, katalog na życzenie),

— dyskietki dołączone do ST$^E$fana oznaczone są literą S oraz numerem i datą wydania czasopisma (dyskietki te nie są uwzględnione w spisie).

W wypadku, gdy istnieją szczególne wymagania co do sprzętu, zostało to podane w nawiasie po opisie programu:

— litery N, Ś, W oznaczają odpowiednio niską, średnią i wysoką rozdzielczość,

— litera E oznacza, że program przeznaczony jest dla Atari ST$^E$,

— cyfra określa minimalną wielkość pamięci, o ile przekracza ona 1 Mb.

Większość programów została skompresowana i dyskietka zawiera kilka/kilkanaście programów, które do tej pory były dostępne oddzielnie. Skompresowane programy są plikami o rozszerzeniu TOS, które należy przekopiować (każdy osobno!) na czyste dyskietki i uruchomić w celu rozpakowania.

Niestety zmiany te, jak również wprowadzenie VAT-u spowodowało zmianę cen dyskietek z naszego zbioru. Obecnie wynoszą one:

— przy płatności za zaliczeniem 35 000 zł za dysk plus przy zakupie poniżej 10 szt. koszt wysyłki 10000 zł.

— przy płatności na konto 30 000 za dysk plus przy zakupie poniżej 10 szt. koszt wysyłki 10000 zł.

Termin realizacji zamówienia od 14 do 21 dni. W wypadku gdy nie jest możliwa wysyłka którejś z zamówionych dyskietek, zamówienie jest realizowane bez niej, a płatność oczywiście zwracana.

Uwaga!!! **Ceny podane na kartach zamówieniowych są nieaktualne z dniem 1.09.93!**

W wypadku jeżeli złożyliście Państwo zamówienie po 1.09.93 lub do 15.09.93 nie otrzymacie Państwo dyskietek prosimy o ponowienie zamówienia według nowych zasad. Za wszelkie niedogodności, opóźnienia oraz nie zrealizowane zamówienia przepraszamy.

Redakcja

**Programy narzędziowe**

*Dysk N/001*

UNITERMINAL — jeden z najlepszych programów komunikacyjnych. (Ś,W)

FCOPY 3.0 — najpopularniejszy program kopiujący. (Ś,W)

SUPERFORMATER — program formatujący dyski w podwyższonej gęstości. (Ś,W)

TIME ACC — zegar i kalendarz w akcesorium. (Ś,W)

TERM ST — program komunikacyjny. (Ś,W)

**Programy edukacyjne**

*Dysk E/001*

QUIZ — edukacja, program testujący wiadomości. (W)

PAUK — edukacja, „dzięcioł", program uczący. (W)

HYPERVOC — edukacja, słownik angielsko-niemiecki (inne języki po dokonaniu korekt). (W)

MACHINACH — gra ucząca zapamiętywania i powtarzania za komputerem ruchów; 4 stopnie trudności. (W)

OWARI — gra logiczna. (W)

SPIROGRA — rysowanie elips i innych figur owalnych. (W)

EUROQUIZ — test ze znajomości geografii, historii, polityki gospodarki i sztuki Europy.

ECS, VOCMAN — programy wspomagające naukę języka angielskiego, uczą odmiany i czasowników nieregularnych. (W)

CARPET, DREI D, FUNKTION, PLOTTER — programy generujące wykresy funkcji. (W)

**Programowanie (języki programy narzędziowe i przykłady)**

*Dysk P/001–P/005*

GNU C++ — znakomity pakiet dla programistów znających C, zawiera edytor, kompilator, asembler, debugger oraz programy pomocnicze i biblioteki. Niestety pakiet ma spore wymagania sprzętowe, (najlepiej dysk twardy i 2MB RAM) i niewygodny interfejs użytkownika. (Ś,W,2)

*Dysk P/006*

FORTH — język programowania. (W)

**Programy użytkowe**

*Dysk U/001*

ZEITMANAGER — organizator czasu. (W)

ST DESIGN — program graficzny. (W)

FONTEDIT — edytor fontów wektorowych do ST DESIGN. (W)

CHARTS i DYNA — graficzny zapis kursów papierów wartościowych. (W)

KALENDER — kalendarz. (W)

*Dysk U/002*

EBMATAW — instrukcja ukazująca jak działa XEDIT. (W)

XEDIT — procesor tekstu. (W)

IDEALIST 3.0 — program łamiący i przygotowujący tekst do druku. (Ś,W)

BIORYTMY — szereg programów do generowania wykresów biorytmów, również biorytmy partnerskie, wykresy wypadkowe, itp. (W)

AUDIO 2000 — program do drukowania nalepek na CC, DAT, CD, LP. (W)

CP6-ETIKET — program do drukowania nalepek na dyski 3,5", przy użyciu NEC CP6, druk kolorowy! (Ś,W)

FAST LIFE — symulacja rozwoju populacji (pole 640 x 400). (W)

LIFE IS LIFE — mniejsze pole robocze, ale wiele innych możliwości. (W)

*Dysk U/003*

ADISORT — program pomocniczy dla użytkowników bazy danych ADIMENS. (W)

CARDFILE — kartotekowa baza danych, może służyć jako notatnik, spis telefonów, adresów itp. (W)

*Dysk U/004*

CALAMUS SL — demo wersja, bez komentarza... (W,2)

*Dysk U/005*

PAKIET ANTYWIRUSOWY — ten dysk zawiera

zestaw programów przydatnych każdemu posiadaczowi ST, pozwalających zabić wirusy. (Ś,W)

*Dysk U/006*

DRINK BAR — baza danych dla... barmana (dziesiątki drinków!). (W)

VIENNA — baza danych dotycząca Wiednia, plan miasta, plan metra, 130 najważniejszych budynków, ok. 140 ulic, odległościomierz. (W)*Dysk U/007*

ZX 81 EMULATOR + 65 gier — emulator legendarnego już dziś SINCLAIR ZX 81. (Ś,W)

**Programy muzyczne**

*Dysk M/001*

STE CONNEX STEREO oraz moduły ALF, HARLEY M, J.M. JARRE, JUNGLE, MAGNETIC, RIPPED, SHIT — program do odtwarzania modułów muzycznych z Amigi wraz z modułami. (Ś,E)

*Dysk M/002*

AUDIO SCULPTURE oraz HARLEY, J.M, JUNGLE, KREM, MIX, MUSIC, RAZOR, RIP, VIVEMOI 2 — sound tracker na Atari ST; obsługuje niestandardowe wyjścia centronics, catridge, dodatkowo 8 modułów muzycznych, program kompatybilny z przystawką STEREO do Atari ST. (N)

*Dysk M/003*

ST CONNEX oraz DAWN, KREM, MIX, RAZOR, RSI — player do modułów wersja na ST,dodatkowo 5 utworów. (N)

*Dysk M/004*

AMEGAS, AXEL F, CREMONA, DRAG, DRAGON2, KUTREMIX — moduły do odtwarzania przez np.: ST/E CONNEX, AUDIO SCULPTURE, itp.

*Dysk M/005*

BLUEMONDAY, ORPHEUS, PARALA, POINT, REVOLUTION, SALWIK, SMOKE, SPACENA — jak wyżej

*Dysk M/006*

MIKOŁAJ, LOADER, PIANO, RAZOR, ROXETTE, SEQUENCE, STARDUST, TIGER — moduły do dowolnego soundtrackera.

*Dysk M/007*

BRAINY, FAN3, FAN5, FULCON, GALAXY, IT'S, JAPAN — moduły

*Dysk M/008*

BSTARS, DSTRIKE, EXTASY, FAN, FAN2 — moduły do programu soundtracker

*Dysk M/009*

2, 3, AXEL, ADAMSKI, AFTER, GALAXY, LETSWING — moduły

*Dysk M/010*

EGYPTIAN, IT DREAM, JAPAN, KOTO VIS, LATOROSL, NEW AGE — moduły

*Dysk M/011*

BEAT & SOUND MACHINE — edytor muzyki i automat perkusyjny, pracuje na zdigitalizowanych samplach, stąd wysoka jakość dźwięku, wyjście na przetwornik D/A. (W)

SMS SONG EDITOR — edytor muzyczny, podobny do sountrackera, ale pracuje w innym formacie, jego zaletą jest bardzo dobra jakość dźwięku. (W,Ś)

*Dysk M/012*

MERLIN — program muzyczny shareware, niesamowite możliwości: sequencer, sampler, obsługa centronicsa, MIDI, cartridge, muzyka analogowa i cyfrowa, edycja dźwięku. (W,Ś)

**Programy graficzne**

*Dysk G/001*

NEOCHROME — program malarski i przykładowe rysunki. (N)

MICROCAD — program rysunkowy. (W)

AIM — Atari IMage processor — program do cyfrowej obróbki obrazu — filtrowanie, zmiana składowych, cieniowanie, konturowanie, edycja ziarnistości i wiele, wiele innych możliwości. (Ś,W)

*Dysk G/002*

AMOK SW, GSERVER, MASTERPAINTER — zestaw programów graficznych, konwersje na różne formaty, kompresja i dekompresja. (W)

FINE LINE, C_CHROME — programy graficzne na niską rozdzielczość. (N)

*Dysk G/003*

FUN FACE — portret pamięciowy z komputera, oprócz gotowych wzorów możliwość retuszu, wspaniała zabawa, spróbuj narysować znajomego! (Ś,W)

DIPS — program do cyfrowej obróbki zdjęć. (Ś,W)

**Gry (monitor kolorowy)**

*Dysk GK/001*

LAZERBALL — gra zręcznościowa, twoim zadaniem jest odbijać promienie lasera, musisz uważać, aby nie zniszczyć swojej bazy.

TRADER — jako przywódca kosmicznego imperium musisz obronić Ziemię przed najeźdźcami. Całkiem niezła gra strategiczna podobna do EMPIRE.

DRACHEN — ciekawa gra zręcznościowa.

SUPER GAME — OLIMPIADA, czyli najlepszy sposób na połamanie joysticka.

CODENAME: BOMB, PYRAMID — dwie świetne gry zręcznościowe napisane w STOS Basic (najlepszy przykład na to, co można wycisnąć z tego języka).

*Dysk GK/002*

WAR JEEP — gra zręcznościowa, jeździsz i walczysz amfibią. Niezła grafika (oczywiście STOS Basic).

YPSILON — jesteś głodnym, długim wężem, zdobywając pożywienie musisz uważać, aby nie ugryźć się w ogon.

SPEDEMO — zręcznościówka z niezłą grafiką i animacją.

ARGON — gra zręcznościowa, latasz pojazdem po labiryncie.

DISCHUNT — szukasz dyskietek (3,5'', oczywiście) w ogromnym i pełnym niebezpieczeństw labiryncie.

OUT — musisz wydostać się z labiryntu, w tym celu zbierasz skarby i walczysz z potworami.

TURBO — czyli pasjonujące wyścigi samochodowe dla dwóch graczy, dobra grafika.

*Dysk GK/003*

OXYD COLOR — stary przebój w kolorach to zupełnie inna zabawa

*Dysk GK/004*

ROSEMARY — papier, kamień, nożyczki, stawką bielizna trzech pań.

XENON II — fantastyczna strzelanina, lecz w wersji demo tylko 2 etapy.

*Dysk GK/005*

STARTREK — wszystko o znanym serialu telewizyjnym

*Dysk GK/006*

CHRONICLES OF OMEGA — pierwszy poziom przebojowej gry firmy ARC.

**Gry (monitor monochromatyczny)**

*Dysk GM/001*

HASCS II — gra strategiczno-militarna.

SPACE INVADERS — kosmiczni najeźdźcy i Ty, bohaterski obrońca.

ANDURIL — podróże aniołka po tajemniczym labiryncie.

DIAMOND — klon Boulder Dash'a.

FOOTBALL MANAGER — gra strategiczna, opieka nad drużyną piłkarską, transfery, finanse.

WIZZY — głodna dżdżownica... uważaj, by nie zjeść swojego ogona.

*Dysk GM/002*

EUROQUIZ — test ze znajomości geografii, historii, polityki gospodarki i sztuki Europy.

TEMPTEST — gra zręcznościowa oparta na zasadzie automatu kasynowego „jednoręki bandyta".

TAKE TWO — starochińska gra polegająca na łączeniu odpowiednim systemem dwóch identycznych szablonów wśród układanki (super).

CRAZYWAY — gra przeznaczona dla dwóch graczy; chodzisz po labiryncie i musisz zdążyć przed partnerem zebrać litery składające się na tajemniczy wyraz; niestety wyrazy te są w języku niemieckim.

FUTURE GARDEN — jesteś kulką-ogrodnikiem i pomimo wszelkich przeszkód musisz siać kapustę.

PUZZORY — z kilkudziesięciu elementów musisz ułożyć mapę świata, niemiecki plakat, itd... w ciągu danego przedziału czasu.

*Dysk GM/003*

OXYD — gra logiczna polegająca na włączeniu odpowiednich identycznych wzorów za pomocą latającej kulki; przebój w Niemczech... tysiące sprzedanych egzemplarzy!

*Dysk GM/004*

OXYD 1 — kolejna wersja, umożliwiająca przejście następnych etapów.

*Dysk GM/005*

OXYD 2 — nowe plansze, stare zasady.

*Dysk GM/006*

SPACOLA — strzelanka z doskonałą muzyką, pokrewieństwo do OXYD-a (ten sam autor) gwarancją świetnej zabawy.

*Dysk GM/007*

KRABAT — szachy.

PUZZLE — ciekawa koncepcja i dobra realizacja.

BALLERBURG — ostrzał artyleryjski zamków bawiąc, uczy balistyki.

PATIENCE — dla przesądnych pasjans.

HUANG SHI — starochińska gra logiczna, wykonanie jak na Macintoshu.

MINI MIND — popularna niegdyś gra w odgadywanie położenia pionów i kolorów.

GERHILDS CHALLANGE — gra o fabule zbliżonej do bilardu lub golfa.

**Programy demonstracyjne (monitor kolorowy)**

*Dysk D/001*

FOREIGN AFFAIR — zdigitalizowany utwór Mike'a Oldfielda.

*Dysk D/002*

STAR WARS — demo; zdigitalizowane fragmenty filmu, ponad 1 minuta dźwięku i 150 obrazów!

*Dysk D/003*

COME ON, COME ON — digitalizacja utworu Bronskie Beat.

*Dysk D/004*

SNOWMAN — jeden z najpiękniejszych programów demonstracyjnych, wzruszająca historia przyjaźni chłopca i bałwanka.

*Dysk D/005*

INTRO CONCEPT.

*Dysk D/006*

ULM MEGADEMO — „Dark side of the spoon".

*Dysk D/007*

BRAIN DAMAGE dysk 1 — aktualnie (1.09.93) najlepsze demo na ST^E^. Prod. grupy Agression. (E)

*Dysk D/008*

BRAIN DAMAGE dysk 2 — druga dyskietka dema.

❑

↗15

0, 1 — obliczane są tylko kolory

2, 3 — dodatkowo uwzględniane są wartości dotyczące rodzaju powierzchni

4, 5 — j.w. + cienie

6, 7 — j.w. + textury

8, 9 — j.w. + odbicia, załamania

–l‹path› — scieżka dostępu do katalogu z biblioteką Opcja ta służy do ustawienia katalogu, w którym mają być poszukiwane pliki zawierające np.: definicje objektów, tekstur, kolorów, itp. Tam też będą poszukiwane zbiory zawierające grafikę do nakładania na obiekty (ang. *image mapping*). Możliwe jest podanie wielu ścieżek dostępu, innych dla każdego rodzaju zbiorów. Dokonuje się tego poprzez wielokrotne użycie opcji +l‹path›, np.:

+lINCLUDE +lGRAPHICS +lE:\RASTROWA\GIF +lE:\RASTROWA

W ten sposób można zdefiniować ponad 10 rożnych ścieżek dostepu do katalogów. Standardowo POV-Ray poszukuje danych w katalogu bieżącym, więc nie trzeba tej ścieżki podawać w linii komend.

Spośród tych opcji obowiązkowo musi być podana nazwa pliku wejściowego oraz rozdzielczość (pionowa i pozioma), np.:

POV68000 +iexample.pov +w100 +h80

W tym przypadku obliczona scena zawarta w pliku EXAPLE.POV zostanie zapisana w pliku DATA.TGA w formacie targa, zaś pliki biblioteczne będą poszukiwane w bieżącym katalogu.

Pełna wymagana i zaleca linia parametrów powinna mieć postać (w naszym przykładzie):

POV68000 +iEXAMPLE.POV +oEXAMPLE.TGA +w100 +h80 +v +b30 +p +q9 +lINCLUDE\

Ponieważ wklepywanie za każdym razem takiej ilości parametrów jest nieco pracochłonne, można sobie uprościć życie tworząc plik tekstowy (proponuję rozszerzenie .DEF) zawierający wszystkie parametry. Teraz nasz przykład będzie wyglądał tak:

POV68000 EXAMPLE.DEF

lub jeżeli w pliku EXAMPLE.DEF nie podamy nazwy pliku wejściowego oraz wyjściowego:

POV68000 EXAMPLE.DEF +iEXAMPLE.POV +oEXAMPLE.TGA

W tym przypadku program POVRAY wywołujemy z jednym tylko parametrem tzn. nazwą pliku opcji. Można także (co ja osobiście zrobiłem) zaistalować używaną, w zależności od konfiguracji, wersję POV'a jako aplikację wywoływaną przy podwójnym naciśnięciu przycisku myszki na ikonie z rozszerzeniem .DEF z jej nazwą jako parametr. To powinno uprościć do minimum procedurę uruchamiania programu. Można także podać opcje bezpośrednio w tekście źródłowym za pomocą rozkazu, np.:

SET POVRAYOPT = +v +b30 +ft +w160 +h120 +a0.3

Jeżeli za pomocą tych trzech sposobów podawania opcji ustalimy parametry, które są ze sobą sprzeczne, np. podamy różne rozmiary rastra wyjściowego to uznane za obowiązujące rozkazy podane bezpośrednio w linii komend, następnie w pliku z opcjami a na końcu parametry rozkazu SET POVRAYOPT.

Na tym chciałbym zakończyć pierwszy odcinek. W następnym podam nieco informacji dotyczących wykorzystania bibliotek, ustawiania kamery i oświetlenia oraz podstawowych obiektów i ich transformacji. A więc do następnego artykułu...

### Literatura


1. Jankowski M.: Elementy grafiki komputerowej. Warszawa, WNT 1990, s.212–235.
2. Szczeniowski Sz.: Fizyka doświadczalna. Wyd 5, Cz. 4, Warszawa, PWN 1983.




# Dyskoteka ST$^{E}$fana

## DYSK 5/93:

**AUTOMATY**

AUTOM1D.C
KOMORKA.C
KOMORKA.H
AUTOM1D.PRG
AUTOM1D.PRJ

Kody źródłowe i program wykonywalny do artykułu o automatach komórkowych ze strony 29.

**FX45**

FX45.GFA
FX45.INL
FX45.PRG
FX45.S
PIC.SPU

Kody źródłowe, program wykonywalny i dane do artykułu „FX45" ze strony 28.

**POV**

POV68000.TTP

Raytracer na procesor 68000 (lub wyższe) — patrz artykuł na stronie 14.

**TIME_OPT**

1
2
3
TIME_OPT.DAT
TIME_OPT.DOC
TIME_OPT.PRG
ST_SPEECH.TOS
SPEAK.TTP

Program nadesłany przez naszego Czytelnika, Krzysztofa Kiziaka z Poznania. Służy do wyliczania optymalnego czasu i rozłożenia utworów na kasecie o podanej długości.

**TOTO**

FONTS\ MONACO06.FNT
MONACO08.FNT
MONACO16.FNT
MONACO32.FNT
RSC\ TOTO.HRD
TOTO.RSC
TOTO.PRG

Demonstracyjna wersja programu naszego redakcyjnego kolegi Darka. Wspomaga on wielbicieli toto lotka.

**PHOENIX.(+)**

PHOENIX.DOC
PHOENIX.PRG

Kolejny program Krzyśka Kiziaka. Pozwala on na proste operacje na dysku, w szczególności informuje o formacie dyskietki, rozmieszczeniu poszczególnych plików na dyskietce itp.

## DYSK 6/93:

**INKAUST**

INK.SET

Poprawiony plik konfiguracyjny do demonstracyjnej wersji Inkaustu 2.1 z naszej poprzedniej dyskietki (4/93). Patrz sprostowanie na stronie 27.

**POV**

POV68030.TTP
POV68881.TTP
POVTT.TTP

Ciąg dalszy POV-a. Są to kolejno: wersja na procesor 030, na koprocesor, oraz na TT (czyli 68030+68881).

**Listingi**

Listingi do artykułów: o emulatorze VT-52 (strona 10) oraz „Szara skrzynka" (strona 18).



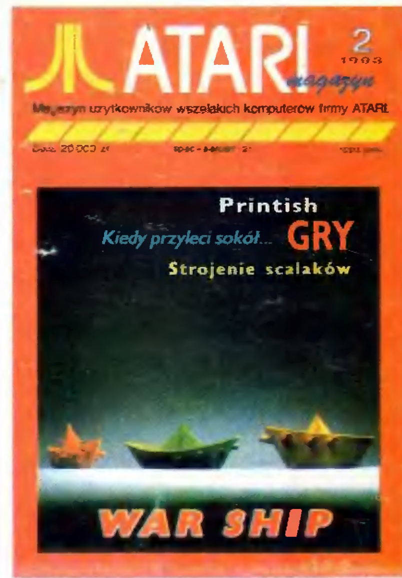
# MAGAZYN DLA CIEBIE

## a w nim:

- **hardware** *(opisy rozwiązań sprzętowych – „przeróbki", itd... – oraz testy sprzętu)*
- **software** *(opisy programów, zarówno tych najnowszych, jak i najciekawszych ze „staroci")*
- **kursy** *(nauka programowania i obsługo komputerów, wraz z opisami produktów do tego używanych – języki programowania, programy graficzne, muzyczne, MIDI, kalkulacyjne. DTP...)*
- **porady techniczne** *dla użytkowników wszystkich komputerów Atari*
- **gry** *(w każdym numerze opisy kilku najnowszych i najciekawszych gier)*
- **inne** *(felietony, wywiady, reportaże, polemika z Czytelnikami...)*

*Jednym zdaniem:*
***WSTYSTKO, CO CIĘ MOŻE ZAINTERESOWAĆ!!!***




# ATARI

XL • XE • ST • STE • Mega ST • Mega STE • TT • Falcon 030

3

ATARI magazyn
Magazyn użytkowników wszelakich komputerów firmy ATARI.
INDEKS 321850
wrzesień - październik (3)
Cena 20.000 zł
Sampler dla maluchów
Asembler dla starszych
Cubase
GRY
EDYTORY
SOKOLE OPOWIEŚCI

Już w kioskach !